



On-line Help

Version 5.0 (October 26, 2000)

CGTech on-line Help works best with **Internet Explorer 5.0** or higher, or **Netscape 4.75** or higher.

VERICUT simulates or replaces costly steps of the NC data verification process, plus optimizes the efficiency of material removal.

<= Use Contents/Index/Search tabs, or click on a topic below:



What's **NEW** in 5.0

Introduction to VERICUT

VERICUT Functions Reference / Map of 4.4 functions in 5.0

CGTech On-line Help Library

Tip: Press "F1" in any VERICUT window to receive help on that window.

Copyright information

Copyright © 2000 CGTech. All Rights Reserved. User documentation from CGTech is subject to copyright laws of the United States and other countries and is provided under a license agreement, which restricts copying, disclosure, and use of such documentation. CGTech hereby grants to the licensed user the right to make copies in printed form of CGTech user documentation provided on software or documentation media, but only for internal, noncommercial use by the licensed user in accordance with the license agreement under which the applicable software and documentation are licensed. Any copy made hereunder shall include the CGTech copyright notice and any other proprietary notice provided by CGTech. User documentation may not be disclosed, transferred, or modified without the prior written consent of CGTech and no authorization is granted to make copies for such purposes.

Information described in this document is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by CGTech. CGTech assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

Registered Trademarks of CGTech: VERICUT, OptiPath, and RevPost.

Trademarks of CGTech: AUTO-DIFF, CATV, FastMill, PolyFix, and X-Caliper.

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

Table of Contents

<i>The Biggest Changes in 5.0</i>	7
<i>Release Details</i>	8
Chapter 1 Introduction	14
<i>Introduction to VERICUT</i>	14
<i>About CGTech documentation</i>	16
<i>Coordinate systems and angles</i>	16
<i>Licenses and options</i>	18
Chapter 2 Using VERICUT	31
<i>Overview of using VERICUT</i>	31
<i>VERICUT coordinate systems</i>	31
<i>VERICUT windows</i>	34
<i>Working with VERICUT files</i>	42
<i>Working with views</i>	57
<i>Using colors</i>	76
<i>Modeling in VERICUT</i>	85
<i>Integrating CAD/CAM data</i>	103
<i>Configuring for tool path simulation</i>	116
<i>Describing cutting tools</i>	157
<i>Controlling and monitoring the simulation</i>	173
<i>Analyzing models and machines</i>	177
<i>Capturing VERICUT images</i>	185
<i>Machine and control building</i>	190
<i>Optimizing tool paths</i>	221
Chapter 3 Reference	222
<i>Map of 4.4 functions in 5.0</i>	222
<i>Machine Simulation 4.4 features => VERICUT 5.0</i>	225
<i>VERICUT Functions Reference</i>	229
<i>G-Code Settings "WireEDM" tab features</i>	323
<i>Find Word/Address window features</i>	359
<i>Tables Reference</i>	417
<i>Control Macros and Functions Reference</i>	437

Help on VERICUT

<i>Command Line Options & Batch Processing</i>	467
<i><uservariablename>=<uservalue></i>	468
<i>addusr=<addusrfile></i>	468
<i>append_log</i>	469
<i>auto-diff=<reportfilename></i>	469
<i>batch</i>	469
<i>batchp</i>	470
<i>datalink</i>	470
<i>export</i>	470
<i>fitall</i>	470
<i>IMRCAPDT=on</i>	470
<i>ip=<ipfile></i>	470
<i>LGFILNAM=<logfilefilename></i>	471
<i>optimize</i>	471
<i>record</i>	471
<i>reset_cutcolor</i>	472
<i>reset_log</i>	472
<i>results</i>	472
<i>reverse</i>	472
<i>usr=<usrfile></i>	472
<i>zoom_out[=<value>]</i>	473
<i>Obsolete command line options</i>	473
<i>IGES Converter command line options</i>	474
<i>Mastercam NCI Converter command line options</i>	475
<i>PolyFix Converter command line options</i>	475
<i>Surface to Solid Converter command line options</i>	476
<i>VDA Converter command line options</i>	476
<i>Environment variables</i>	479
Chapter 4 VERICUT Files	481
<i>File Types and Descriptions</i>	481
<i>Add-user file</i>	482
<i>APT Output file</i>	483
<i>APT Table file</i>	484
<i>AUTO-DIFF Report file</i>	485
<i>CGTech Macro Executable file</i>	486
<i>Control file</i>	486

Help on VERICUT

<i>Control Report file</i>	487
<i>Design Points file</i>	488
<i>G-Code Log file</i>	489
<i>G-Code Report file</i>	490
<i>IGES model file</i>	491
<i>Image file</i>	491
<i>In Process file</i>	492
<i>Job file</i>	492
<i>Log file</i>	492
<i>Machine file</i>	494
<i>Optimized file</i>	495
<i>OptiPath Library file</i>	495
<i>Preferences file</i>	495
<i>Results file</i>	496
<i>Status file</i>	498
<i>Stereolithography model file</i>	498
<i>Stereolithography model file</i>	499
<i>Tool Library file</i>	499
<i>Toolpath file</i>	500
<i>User file</i>	501
<i>VERICUT Linear-sweep file</i>	502
<i>VERICUT Polygon file</i>	503
<i>VERICUT Solid-of-revolution file</i>	505
<i>Supported APT-CLS tool path records</i>	508
<i>Control files</i>	528
<i>Machine files</i>	532
<i>Sample-demo files</i>	535
Chapter 5 Training Sessions	536
<i>Session 1- Introduction to the VERICUT user interface</i>	536
<i>Session 2- Modeling a casting & fixture</i>	544
<i>Session 3- Modeling and assembling models</i>	554
<i>Session 4- Cut multiple setups using the same tool path</i>	575
<i>Session 5- Simulate multiple tool path files, same orientation</i>	579
<i>Session 6- Simulate multiple tool paths, different orientations</i>	583
<i>Session 7- Create a Tool Library</i>	594
<i>Session 8- Mill tool profiles and a default tool</i>	597

Help on VERICUT

<i>Session 9- Use a tool list and "master" Tool Library</i>	608
<i>Session 10- Visually analyze a cut model</i>	615
<i>Session 11- Measure a VERICUT Model Using X-Caliper</i>	619
<i>Session 12- AUTO-DIFF and Comparing a Design Model</i>	622
<i>Session 13- Control the Tool Path Simulation</i>	629
<i>Session 14- Command Line Options & Batch Processing</i>	633
<i>Session 15- Model Export</i>	637
<i>Session 16- Configure a User file to Simulate G-codes</i>	639
<i>Session 17- Use OptiPath Manager to Create an OptiPath Library</i>	642
<i>Session 18- Optimize Tool Path Feedrates via Tool Library method</i>	647
<i>Session 19- Optimize Tool Path Feedrates via OptiPath tool list method</i>	651
<i>Session 20- Build/Modify an OptiPath library while building a tool list</i>	654
<i>Session 21- Run OptiPath interactively</i>	657
<i>Session 22- Build a Kinematic Model of a 3-axis Mill</i>	660
<i>Session 23- Build a Kinematic Model of a 4-axis Horizontal Mill</i>	667
Chapter 6 ERRORS and Problem Solving	674
<i>Error messages</i>	674
<i>Error conditions</i>	681



What's NEW in 5.0

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

October 24, 2000

Dear VERICUT® User:

Congratulations on your purchase of CGTech software - your most valuable digital manufacturing tool for improving NC productivity!

The 5.0 release contains a new generation of NC simulation, verification, and optimization technology-packed with features that make VERICUT more powerful and easier to use. This letter describes the important changes in VERICUT 5.0. Take a moment to review what's new and improved in this release.

VERICUT 5.0 supports milling machine operations. Look for new NC turning and mill-turn solutions in the next release- version 5.1.

System requirements have changed - We recommend that you carefully read the system requirements and all installation instructions that came with your CD before loading any software.

How To Get a License - All users must complete and return the *License Request Form* in the CD booklet, or submit the application at www.cgtech.com/license. Licensing will be sent via Email only. **Note:** Use only 5.0 licensing with this software.

Updates - Note the following changes to the information in the booklet accompanying your CD.

Disk space required to install VERICUT 5.0 => **180 Mb**

Temporary space required => **30-90 Mb**

HP UNIX operating system requirement => **HP-UX 11.0 or later**

UNIX Java Runtime Environment requirement => **JRE 1.2.2.05 (or equivalent- see the *Installation Guide* for details)**

Maintenance and Licensing Information

Your investment in software maintenance keeps you on the cutting edge - CGTech provides upgrade software to customers with current software maintenance contracts. Your continued maintenance ensures that you have the most advanced verification technology available. If your maintenance has expired, please contact your CGTech representative.

Sincerely,

CGTech Technical Support

The Biggest Changes in 5.0

New User Interface

VERICUT 5.0 has a new interface designed to make VERICUT easier to use. Existing users can refer to the "*Map of 4.4 functions in 5.0*" in the *Reference* section to see where favorite 4.4 features went. 5.0 adds many new convenience features, such as: user selectable "look & feel", right-click shortcut menus, and center-click actions. See the on-line Help for details.

Unified VERICUT tool path verification and Machine Simulation

VERICUT now gives you control over what you see in any view. The traditional "Workpiece" view shows the tool removing material on a stationary part. A "Machine" view shows the NC machine. With the Machine Simulation option you see how the machine responds to motion commands, and detect if any collisions occur. A new "Machine/Cut Stock" view shows cutting as it occurs on the NC machine. Right-click to reveal a shortcut menu providing quick access to any of these view types.

Rotate and Cut / Zoom and Cut

A breakthrough in technology enables you to rotate or zoom the workpiece and continue cutting. This same breakthrough makes it possible to do the following at any time: resize the VERICUT main window, add or delete views, and resize views in the VERICUT main window.

Model Export

The new *Model Export* option exports the VERICUT cut model in IGES, STL, or VERICUT model formats, making it possible to bring the as-machined part into your CAD system. Use Model Export to create accurate models for the next machining stage, or from those old NC programs that do not have a CAD/CAM source. You can export sectioned models, or models resulting from comparing models via AUTO-DIFF™.

Speed and Accuracy

Like previous versions, VERICUT 5.0 has consistent robust performance. The 5.0 design processes large or complex tool paths without slowing down during cutting, and reaches previously unobtainable levels of accuracy to support even the most stringent inspection needs.

Easier tool path optimization

The OptiPath® tool path optimization module has many quick and easy ways to setup optimization. Use the OptiPath Manager to see the cutter shape, calculate optimum feeds and speeds for various cutting conditions, then create optimization records based on that data. All OptiPath settings are now centrally located in a single window.

New "prompting" methods of creating optimization records are available at run time, or when you build a tool list. Activated by tool changes, these options open the Optimization Settings window pre-configured with cutter information, waiting for you to describe how to optimize cuts that follow.

Ingersoll OptiPath and Tool libraries

The Ingersoll Cutting Tools company and CGTech are sharing technology to provide users with a unique solution for NC manufacturing. 5.0 contains OptiPath and Tool libraries supplied by Ingersoll Cutting Tools that contain Ingersoll catalogue cutters and optimization values intended to serve as a "starting point" for optimizing cuts in popular workpiece materials, such as: aluminum, cast iron, stainless steel, and much more.

CAD-like Modeling

Modeling the workpiece and machines has never been easier. The "Component Tree" provides a graphical overview of components, models, and how they are related. Copy or delete entire "branches" of components and their models. Translate, rotate, or assemble objects with others. Workpiece setups and machines are modeled the same way. Right-click to reveal a shortcut menu that speeds defining and manipulating components and their models.

Improved AUTO-DIFF

AUTO-DIFF is an option that compares design model data against the as-machined workpiece and detects gouges or excess material. VERICUT 5.0 increases the accuracy of these operations and provides flexible display of the results. Easily switch between solid and surface comparisons, view stock and/or design models, and change inspection tolerances at any time.

New On-line Help System

VERICUT 5.0 comes with a brand new on-line Help system. Press "F1" in any VERICUT window to receive help on that window. **Contents**, **Index**, and word **Search** tabs in a separate pane make information easy to find, without losing your place in the on-line book. Visit the *CGTech On-line Library* for central access to all CGTech Help documents.

Release Details

----- VERICUT Tool Path Simulation -----

Key features of the new VERICUT 5.0 user interface:

New menu arrangement—VERICUT menus are arranged in logical order of use. If you have trouble finding things try the on-line Help **Index** or "*Map of 4.4 functions in 5.0*" in the *Reference* section.

New main VERICUT window layout—In addition to new menus and relocated simulation (VCR) controls, notice the "BUSY"/"OptiPath" lights and progress bar that keep you informed about what's going on. The message area retains recent error and informational messages for quick access.

Enhanced viewing controls—Open as many views as needed to see the cutting process. Each view is individually controlled, capable of displaying the workpiece or NC machine at any time. You can move, resize, tile or cascade, add or delete views at any time. Right-click in a view to access a shortcut menu with options to change view type, background, standard views, and more.

New backgrounds—Options are available to display a shaded background, or walls/floor/ceiling (machine views only).

Look & Feel—Change the entire look of the user interface with a single click (**View>Look & Feel**). The current look & feel is used the next time you run VERICUT.

More OptiPath improvements

Consistent optimization—Optimization in 5.0 produces more accurate feed rates with less oscillation. Optimization is not affected by number of views, view size, or orientation.

Interactive optimization—No pre-setup required! Now you can setup to optimize tool paths via interactive prompting at runtime or when you build a tool list. Click "**Interactive**" on the

OptiPath>Control window to have VERICUT prompt for optimization settings when the tool is changed, or choose the related option on the Tool Change List window to be prompted when building a tool list. Both methods automatically save optimization settings in an OptiPath Library file.

Technology designed for ease of use AND power—The OptiPath Manager has a sleek new design that puts all optimization settings at your fingertips in one window. Sliders make it easy to enter feed/speed information. An interactive cutter graphic shows the cutting condition for which optimization values are being entered. You can instantly sort optimization records by clicking on table headings.

Combined chip thickness and volume removal optimization—With both of these options selected on the **Feed/Speed** tab, VERICUT considers optimization values calculated by each method and uses the one that is more conservative.

Angle adjustment works with all methods—Previously available only with feed and depth table optimization, you can now use the table on the **Angle** tab to affect cuts optimized by any method.

Default values for many settings—When optimizing G-code tool paths, VERICUT establishes defaults values from the NC machine configuration for the **Air Cut Feed Rate** and other values located on the **Settings** tab. Use the **Default** values or clear the checkboxes and enter new values.

Change optimization values in the middle of a cut—Users that change cutting conditions drastically in a single cut sequence can now reference different optimization records via comment records in the tool path file. See "PPRINT/VERICUT-OPTIPATH" under the section entitled "Supported APT-CLS tool path records" in the *Reference* chapter for details.

More status on optimization—Optimization results are at your fingertips. The **Info>Status** window Tool Use Graph and optimized time show you the results of optimizing your tool paths. A new Results file summarizes optimization results and provides links to the optimized tool path file and VERICUT Log file.

More AUTO-DIFF improvements

Flexible display—Added ability to control which models are seen in the comparison results: stock, design, or both. You can also see gouges, excess material, or both resulting from any comparison.

AUTO-DIFF tolerances and colors—Specify tolerances and corresponding error colors directly in the AUTO-DIFF window. Surface comparison provides multiple colors and tolerances that produce a "rainbow" affect to show the extent of errors.

Error measurement highlights—Click on **Report** to open the AUTO-DIFF Report window containing an interactive list of errors detected by the current comparison operation. Click on any item in the list and VERICUT highlights the "measure from" point (+) and the "measure to" point (o). Choose **File>Save** in the AUTO-DIFF Report window to save the inspection report to a file.

Multiple AUTO-DIFF operations—You can perform multiple AUTO-DIFF comparisons in 5.0 without restoring the original cut model. VERICUT automatically restores the cut model for you for the next comparison.

Manipulate AUTO-DIFF results—The AUTO-DIFF comparison model can be rotated, zoomed, panned, saved as an IP file, or exported as an IGES or STL model (see "Model Export" above).

Modeling

Flexibility and power—A new **Model>Model Definition** window is designed to handle many tasks, such as: defining and moving models, moving components or describing their attributes, and importing IGES models. You choose what to manipulate—a model, or the component and all its

Help on VERICUT

models.

Assembly modeling—**Assemble** tab features on the Modeling window can mate or align one to three planar surfaces with surfaces on other models. Graphical selection allows you to "snap" models into place quickly and easily.

Integrated IGES import and surface offset—**Import** tab features make it easy to use IGES data in the simulation as solid models, or design surfaces. You can also create solid models from open surfaces. Apply an offset to "add material" for machining in VERICUT.

Simplified tool path and tool handling

Tool path and tool change activity in one window—A streamlined **Setup>Toolpath** window specifies the tool path(s) for VERICUT to simulate, coordinate systems for each tool path, how to change tools, and more.

Shanks and holders are one—Previous versions of VERICUT supported defining three pieces of a milling tool assembly: cutter, shank, and holder. 5.0 offers two component types: cutter (cutting) and holder (non-cutting). However, multiple holder definitions are available if you prefer all three pieces. To use holders in the simulation select "**Use Holders**" in the **Setup>Toolpath** window.

Tool lists for specialized tool change activity—By default, VERICUT uses tool IDs or pocket numbers to access tools stored in a VERICUT Tool Library. Use tool lists to customize which tool VERICUT uses for each cut sequence, automatically change tools at program stop, skip tool changes, and more. In the list, type the ID tool to use or left-click and choose from an option list. Right-click to add and delete tool list entries. Drag the button at the beginning of an entry to move it in the list.

User-controlled animation speed—Adjust the "**Animation Speed**" slider in the **Setup>Toolpath** window to achieve the desired speed. Select "**Machine Speed Control**" to also affect machine animation.

Easier G-code simulation

VERICUT simulates G-codes that run on most popular NC machining centers. Use functions in the **Setup** menu to choose NC machines and controls from the CGTech library. Open **Setup>G-Code Settings** to enter the information needed by VERICUT to simulate G-code tool paths.

If the machine or control you are looking for is not in the library, try a similar one made by the same manufacturer. If necessary, use **Machine/Control** options in the **Setup** menu to customize existing machines/controls or make new ones.

APT-CLS tool path simulations

Enhanced APT cycle support—VERICUT uses "intelligent pattern matching" to interpret cycle records. Popular cycle formats are simulated with much less cycle data than previous versions. Cycle definitions are grouped and used based on tool path type.

Coordinate systems—Coordinate systems locate an APT or CLS tool path for proper machining. Use **Setup>Coord System** to graphically define, translate, or rotate tool path coordinate systems as needed.

New "Results file"

Click **Info>Results File** to create an HTML formatted Results file overview of the VERICUT session, including: VERICUT file names, auto-saved images, machining times, optimization results, and links to error Log and Optimized tool path files.

Help on VERICUT

Other cool stuff...

"New" VERICUT session without exiting—Click **File>New** to set VERICUT to a "new" condition, as if no files have been loaded.

Automatic file extensions—VERICUT automatically supplies extensions to files being saved when an extension is not specified.

X-Caliper measurement highlights—X-Caliper shows what is being measured by highlighting the "measure from" point (+) and the "measure to" point (o).

MDI motion commands remove material—In addition to moving the NC machine, the 5.0 integration of machine and tool path simulation enables these commands to remove material.

----- **VERICUT Machine Simulation** -----

Machine simulation control—Enabled/disabled via **Setup>Machine>Settings: Simulation On**. To see a 3-D machine you have defined, choose a "Machine" or "Machine/Cut Stock" view type.

Division of modals—Items that were in the 4.4 **Modals** menu are relocated as follows, depending on function. Unlike 4.4, duplicate values can not be saved in both locations.

- Control-related modals => **Setup menu > G-Code > Process Options** (saved in Control file)

- Job-related modals => **Setup menu > Control > Settings** (saved in User file)

Control setup—Use **Setup>Control** menu functions to define words, group words with values and have them perform specific actions (call macros, set variables, etc.), and access control settings and advanced options.

Machine setup—Once machine components are defined (see "Modeling" above), use the **Setup>Machine>Settings** window to set axis priority, travel limits, collision checking, and specify important machine locations.

New "Machine Up" direction—Used by VERICUT to correctly orient the machine relative to the walls/ceiling/floor in a machine view. (Ref. **View>Attributes: Background**.)

New locations for tables—Tables used to store offset values, machine axis locations, and other G-code related data can be found in the following locations in 5.0:

- **Setup>Machine>Settings: Tables** tab—Specify important machine locations and detail tool change motion.

- **Setup>G-Code>Settings: Job Tables** tab—Specify work offsets and machine locations specific to the current tool path file(s).

- **Setup>G-Code>Settings: Tool Tables** tab—Store tool-related offset and compensation data.

----- **New NC Machines and Controls** -----

Following is a list of new NC machines and controls available in the CGTech library and sample files.

Description	Filename	Location
Aerostar H2200 5-axis horizontal	aerh2200.mch	library

Help on VERICUT

mill		
Dixi Jig Center 400 5X FMS	dixjc400.mch	samples
Fanuc "i" series mill controls	fan160im.ctl, fan16im.ctl, fan180im.ctl, fan18im.ctl, fan210im.ctl, fan21im.ctl	library
Haas MiniMill 3-axis vertical mill	hasmini.mch	library
Haas VB-1 5-axis vertical mill	hasvb1.mch	library
Haas CNC control	hascnc.ctl	library
Heidenhain Mill Plus control	heimplus.ctl	library
K&T HB4 4-axis horizontal mill	kt_hb4.mch	library
K&T Gemini E control	gemini_e.ctl	library
Maho DMU 80 P 5-axis horizontal mill	mahdmu80.mch	library
Maho MH700 3-axis horizontal/vertical mill	mah700m.mch	library
Matsuura MAM 600HF vertical mill	mat600hf.mch	library
Okuma OSP 7000M control	osp7000m.ctl	library
Siemens Sinumerik 840C control	sin840c.ctl	library

----- Converters and Interfaces -----

Converters accessible in VERICUT

Convert IGES, Binary CL, Mastercam, and other CAD/CAM data using options in the **File>Convert** menu.

WorkNC Interface on Solaris

VERICUT 5.0 is the initial release with support for the WorkNC Interface to VERICUT on Solaris computers.

Mastercam version 8 support

VERICUT simulates converted NCI data from Mastercam version 8, including drill cycle formats.

----- System-related -----

Windows 2000

Help on VERICUT

Version 5.0 introduces VERICUT for the latest Windows operating system-Windows 2000.

New System Requirements

To support the robust features 5.0 offers, additional memory and dual processors are helpful. We recommend that you carefully read the system requirements and all installation instructions that came with you CD before loading any software.

File input

User files—VERICUT 5.0 supports User files created from VERICUT versions 3.2 or higher.

In Process files— Model data from pre-5.0 IP files is not supported in 5.0. However, you can open the IP file as a User file (via **File>Open**), then re-machine the part to create a 5.0 IP file.

Job files are gone—User and job-related data is combined in the User file. Pre-5.0 Job files can be loaded in VERICUT 5.0 via **File>Open**, but only a User file can be saved.

Graphics devices and accelerators

VERICUT 5.0 performs best with fast "pixmap" graphics devices. Graphics accelerators, such as those using OpenGL, are not effective.

New "Preferences file"

A new Preferences file stores such information as: size and locations of VERICUT windows, file selection filters, names of recently opened User files, and more. This file is automatically updated each time you exit VERICUT.

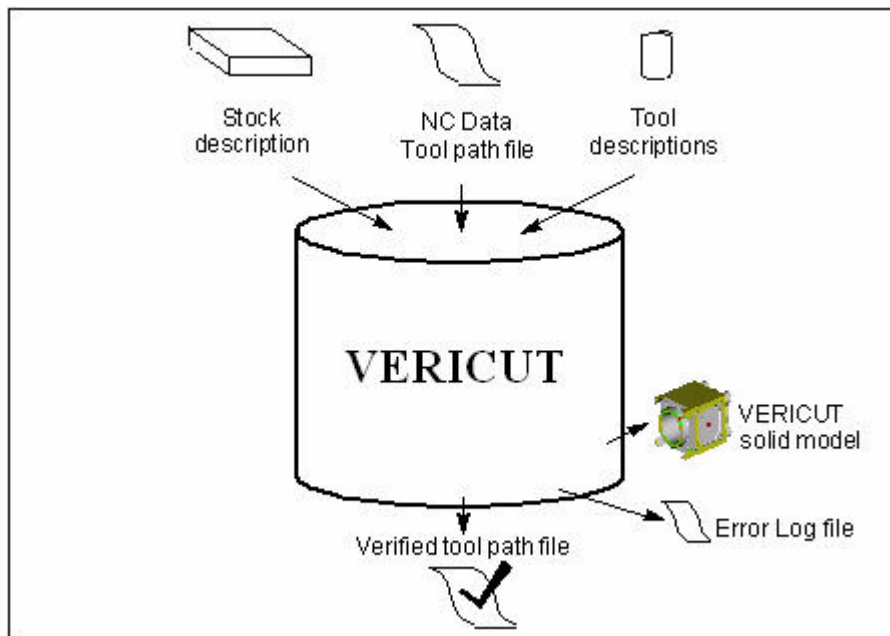
CGTech 9000 Research Drive, Irvine, CA 92618 • www.cgtech.com • Tel: (949) 753-1050 • Fax: (949) 753-1053 • Email: support@cgtech.com

Chapter 1 Introduction

Introduction to VERICUT

VERICUT simulates or replaces costly steps of the NC data verification process, plus optimizes the efficiency of material removal. By simulating NC tool paths, VERICUT can reduce or eliminate the time-consuming and expensive steps of the traditional data prove-outs. The program also optimizes program feed and speeds to increase the efficiency of the machining process. This all adds up to lower costs and higher profits for your business.

Tool path verification process with VERICUT:



VERICUT interactively simulates the material removal process of NC tool path data. Instead of messy wax or foam models, or costly prove-outs that waste valuable production time, a simulated VERICUT part is graphically produced on your computer screen. VERICUT then verifies the accuracy of the tool path and makes certain the finished part matches the design model. Before any machine time is spent, VERICUT catches discrepancies that could corrupt the cutting process and identifies the responsible tool path records for fast and easy adjustment.

The input into VERICUT is tool path data from almost any source. G and M code data as well as APT type CL-files are directly processed by VERICUT. Similar to the requirements of machining a real part, VERICUT needs the tool path data, a description of the raw stock material to be machined, and descriptions of the cutting tools used to machine the part. The result of the verification process is a solid model of the machined part and an error Log file reporting any machining errors detected during the simulation. The VERICUT model can be inspected, saved, or used as the stock material for another tool path.

In addition to streamlining the verification process, VERICUT dramatically boosts productivity on the shop floor. Automated and user-selectable optimization settings in VERICUT let you maximize the material removal process. The feed rate and cutting speed of each tool path automatically adjust based on the depth,

Help on VERICUT

width, and angle of each cut. In fact, you can even set VERICUT to compensate for dull or custom cutters or other elements unique to your operation.

VERICUT simulates up to 5-axis milling, drilling and wire EDM operations, as well as turning and combination mill/turn machining. With a complete line of specialization modules for added features and functionality, VERICUT is truly the total simulation, verification, and optimization solution for unleashing NC productivity.

Accessing VERICUT

The typical method of accessing VERICUT is different, depending on the environment in which it is operated.

Access VERICUT on a Windows computer—From the Taskbar, click **Start > Programs > CGTech VERICUT 5.x > VERICUT** where “5.x” is the desired VERICUT version number.

For example, to run VERICUT 5.0 you would click: **Start > Programs > CGTech VERICUT 5.0 > VERICUT**

Access VERICUT on a UNIX computer—Type “vericut”. If the command is not found, include the full path, e.g. “/ <cgtech5x>/ <computertype>/commands/vericut” where <cgtech5x> represents the directory path to where CGTech products were installed, and <computertype> is the type of computer on which VERICUT is being operated, for example: hp700, rs, sgi, or solaris.

For example, to run VERICUT installed in “/server/apps/cgtech50” on an HP700 computer, you would type: /server/apps/cgtech50/hp700/commands/vericut

Hint: Copy the “vericut” command file to the /usr/bin directory and the next time you log in, you should be able to just type “vericut”.

Access VERICUT from CATIA—Assuming you have the Catia Interface, choose the “CATV” function. Use CATV functions to select the CATIA data to use in the verification session (geometry models, tool paths, etc.) and transfer the data to VERICUT. Then access VERICUT via choosing “**VERICUT**” from the **PROCESS menu**.

Access VERICUT for Pro/Engineer—Access VERICUT via the “CL Data” or “NC Sequence Play Path” menu. The “NC Check” option under the NC Sequence Play Path menu simulates the current selected NC sequence. The “NC Verification” under the CL Data menu enables users to select a single NC sequence, multiple sequences, or an entire NC operation for simulation. **Note:** the NC data must be selected prior to selecting starting the verification process.

Access VERICUT from WorkNC—Access VERICUT via choices in the WorkNC “Utilities” menu. VERICUT can be run interactively or in batch, to perform tool path verification or optimization.

Access VERICUT from UG—Documentation not yet available.

Access VERICUT from Mastercam—Documentation not yet available.

About CGTech documentation

CGTech provides documentation in the following forms:

HTML books

PDF books

All books are available in HTML and PDF formats on the CGTech CD-ROM. You can view HTML books with your Internet browser. To view and print PDF books, you must have the Acrobat Reader installed.

Note: Printed documentation is available from CGTech at extra cost. Please contact your CGTech representative for more information.

CGTech welcomes your suggestions and comments on its documentation. You can send feedback in the following ways:

Send comments electronically to support@cgtech.com, or via the CGTech web site at www.cgtech.com.

Fax comments to CGTech, attention: Documentation Department at (949) 753-1053.

Conventions

CGTech documentation uses the conventions described in this section.

Fonts

The table below describes text fonts used to help orient the user to the CGTech product user interfaces, and actions.

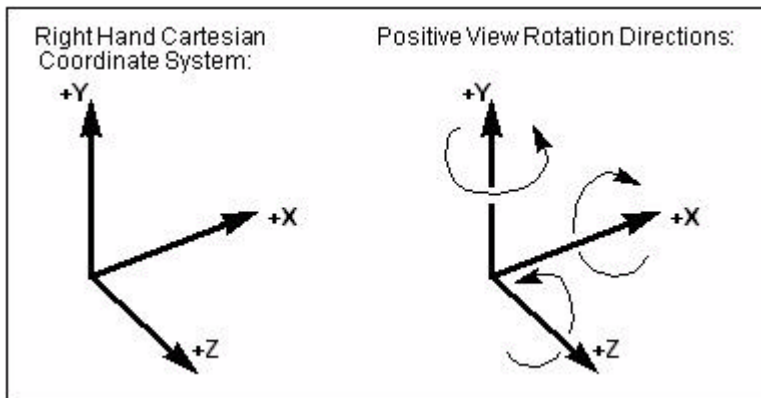
Font type	Where used	Examples
Bold	Menus paths, dialog box options, buttons, and other user interface elements.	Choose File menu > Save . Choose Model Type=Fixture .
<i>Italics</i>	Names of other CGTech documents, system messages, file names.	For more information, consult the <i>VERICUT Installation</i> documentation. When done, you will see a message similar to <i>Processing complete</i> .
Less-than and greater-than symbols (< >)	Variables for which the user substitutes an appropriate value.	Enter File =<yourname.usr> PPRINT/VERICUT-CUTCOLOR <n>

Coordinate systems and angles

CGTech products use a right-hand Cartesian coordinate system to describe location and orientation.

Help on VERICUT

Rotation angle values are specified in decimal degrees. A positive angle goes counter-clockwise about an axis when viewed from the positive end of the axis, looking towards the origin. A negative angle specifies clockwise rotation as shown in the next figure.



Help on VERICUT

Licenses and options

Licensing and VERICUT environments

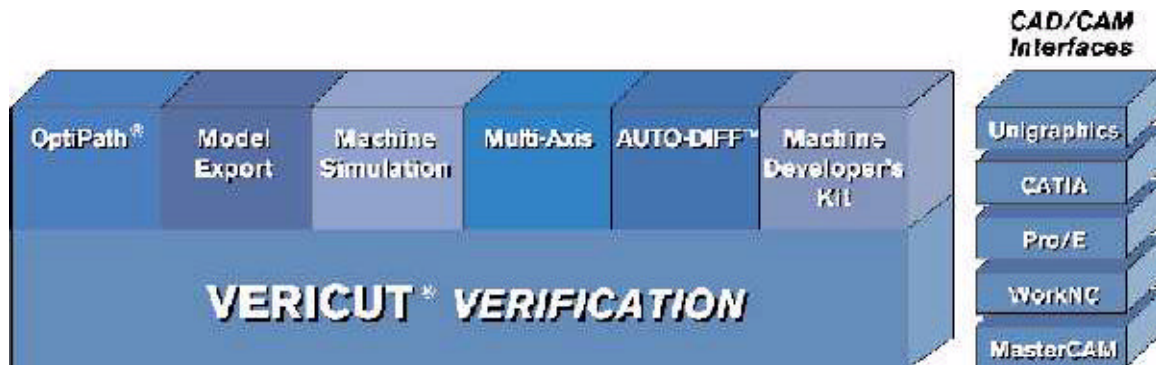
VERICUT licenses and capabilities differ, depending on the environment in which VERICUT is being operated. Consult the appropriate section for details.

Licenses for VERICUT by CGTech

Licenses for VERICUT for Pro/Engineer

Licenses for VERICUT by CGTech

The sections that follow describe the software products obtained from and licensed by CGTech. A modular approach provides the flexibility to purchase only the capabilities you need. As your needs change, you can add the appropriate licenses to increase VERICUT's functionality. It is not necessary to install any additional software to add a license CGTech provides a license that allows immediate access. CGTech software licenses allow you to share VERICUT over a network. One user may access a given license at a time. CGTech's license system supports Windows 95/98, NT, 2000 and UNIX. The server and networked workstations can be different computer types.



Licenses for VERICUT by CGTech:

- Verification
- Multi-Axis
- OptiPath
- Model Export
- Machine Simulation
- Machine Developer's Kit
- AUTO-DIFF
- Catia Interface
- WorkNC Interface
- UG Interface

Verification

The *Verification* license is the foundation of VERICUT. It includes everything necessary to simulate and verify 3-axis milling and multi-axis positioning operations to detect errors such as:

- Inaccurate programming

Help on VERICUT

- Rapid motion contact
- Incorrect tool path motions
- Collisions with fixtures & clamps
- Incorrect or misread blueprints
- Tool shank and holder collisions
- CAD/CAM and post-processor errors

VERICUT has two viewing options: Workpiece, and Machine. The Workpiece view interactively displays material removal and the tool assembly, stock and fixtures. The Machine view displays the CNC machine tool and the complete machining environment, including the cut model as material is removed. On it's own, the Verification license supports the Workpiece view. The Machine view requires a Machine Simulation license.

Model analysis—You can move, zoom, reverse, and rotate the model at any point in the cutting process. You can also cross-section it multiple times at any orientation, and view it in translucent mode enabling you to check areas that would be impossible to see in a solid model (such as the intersection of drilled holes). X-Caliper, a comprehensive measurement tool is included to check thickness, volume, depth, gaps, distances, angles, hole diameters, corner radii, scallop height, and edges. Delta X, Y, Z component distance measurements are included.

Cutter simulation—You can accurately simulate the material removal process using the exact cutter used to machine the part. VERICUT includes a library of standard cutters provided by Ingersoll Cutting Tool Company. You can also define your own cutters. Holders can be specified as a "non-cutting" portion of the tool in order to check for any potential collisions.

VERICUT supports concave bottom, or non-center cutting end mills, such as carbide insert end mills. Although designed to remove large volumes of material at high speeds during planar or XY cutting, carbide insert roughing end mills are often used to ramp into material. If the angle of the cut is too steep, the "non-cutting" portion of the tool collides with the material and can break the tool and ruin the part. VERICUT enables you to define inserts as cutting edges and the body as a non-cutting portion of the tool, helping you take full advantage of your equipment without fear of destroying the workpiece or cutting tool because of an incorrect ramp motion.

FastMill—The FastMill cutting mode is designed to quickly process large NC programs and is especially useful for mold and die manufacturers. A single mouse click switches cutting operations to FastMill mode. FastMill provides complete control of the speed, accuracy, and quality of the model. Accuracy is calculated independently of the display, enabling you to create highly accurate models and display them at lower resolutions for quick manipulation and high resolution for detailed analysis. FastMill uses the same trajectory logic for calculating material removal as the standard cutting mode (e.g. it's built with the same robust simulation technology developed and refined for a decade in VERICUT). FastMill is not a replacement for VERICUT's standard machining mode; it is designed to work with large tool paths at a single setup orientation.

Multi-Axis

As complexity increases, so does the chance for error. Tool positioning is critical in four and five-axis milling operations, further increasing the importance of verification. While it is nearly impossible to predict how these machines will respond to complex motion commands, *Multi-Axis* enables VERICUT to graphically display the results and take the guesswork out of multi-axis tool path prove-outs.

Milling—The *Multi-Axis* license accurately displays material removal during multi-axis milling of complex

Help on VERICUT

profiles and surfaces. The *Multi-Axis* license simulates and verifies four and five axis milling/drilling machining operations.

Turning—The *Multi-Axis* license also simulates and verifies up to 2-4-axis turning operations. VERICUT supports 4-axis synchronized (merged) turning operations with multiple tools.

When combined with Machine Simulation, CGTech's comprehensive 3-D collision-checking can be applied to the in-process workpiece as it is machined. This is useful for detecting interference between the machined part and tool holders, boring bars, tool blocks, turrets, head assemblies, and other machine components, in addition to standard collision checking between machine components.

OptiPath

The *OptiPath* license automatically modifies feed rates based on cutting conditions and amount of material removed. Using *OptiPath* can dramatically reduce the time it takes you to machine parts and increase the productivity (and profitability) of the NC machine tools at your shop.

The feed rate problem—With the current advances in cutting tools, materials, and automatic tool path generation software, it is increasingly critical to use the right feed rate for each and every cut in the NC tool path. However, determining and setting the optimum feed rate has traditionally presented NC programmers and machinists with a number of problems. Because an incorrect estimate can break the cutting tool, damage the fixture, and scrap the part, usually only one or two conservative feed rates are used. Typically, the ‘safe’ feed rates chosen are a compromise between tool life, cycle time and the worst-case cutting condition encountered.

These ‘worst-case’ feed rates are fine for the cuts that remove the largest amount of material or encounter the worst cutting conditions. Unfortunately, these slow, poorly controlled speeds waste time, increase manufacturing costs, and create poor cutting conditions elsewhere on the part.

CAM software does not provide the material removal information necessary to determine optimum feed rates. And many CAM systems have limited cut patterns that do not fit the workpiece geometry. Their machining strategies and cut patterns are not very efficient for common operations such as open boundaries, roughing cuts on complex shapes, complex pocketing, or planar roughing on mold cavities or cores. Cutters therefore waste a lot of time while “cutting air” and feeding slowly across the surface of the part, in contact with it, but not removing any material. Additionally, an automatically generated tool path may plunge the cutter into material at an incorrect or inefficient angle. The end results can include excessive machining time, bad workpiece finishes, increased cutter wear, and broken tools.

Even if you somehow know the optimal feed rates to use for each individual cut, inserting them into the tool path program is time consuming and error prone.

Optimizing the operation—*OptiPath* works by reading the NC tool path file and dividing motion into a number of smaller segments. Where necessary, based on the amount of material removed in each segment, the software assigns the best feed rate for each cutting condition encountered. It then outputs a new tool path, identical to the original one but with improved feed rate settings. It does not alter the tool path trajectory.

This solution offers the best of both worlds. On one hand, it is automatic and determines the best feed rates before the NC program is even loaded on the machine. On the other, it uses the expertise of the NC programmer and machinist to determine the best feed rates for specific cutting conditions. You input ideal feed rates for a number of pre-determined machining conditions. Factors such as machine tool capacity (horse-power, spindle type, rapid traverse speed, coolant, etc.); fixture and clamp rigidity; and cutting tool type (material, design, number of teeth, length, etc.) all affect the process. These factors determine settings (chip thickness, volume removal rate, entry feed rate, spindle speeds, etc.) used to calculate the optimum feed rate for each segment of each cut.

Typically, different optimization techniques are best suited for certain materials or machining processes:

ROUGHING—During planar roughing of aluminum structural components, material is removed at a constant axial depth, but the radial width of cut varies greatly from cut to cut. The goal is to remove as much material as quickly as possible. Maintaining a constant volume removal rate keeps the cutter at its maximum rate of advance into material for the varying cutting conditions. Using the information provided while verifying the tool path, the software is able to determine the amount of material removed in each segment of the tool path. The software then assigns the best feed rate using the information supplied by the NC programmer and/or machine tool operator.

FINISHING—Another example would be semi-finishing or finishing operations in a tool steel mold cavity. Cutting is typically characterized by widely varying chip loads as the tool profiles through the roughing cuts and over the contours of the workpiece to near net shape. In order to achieve a constant chip load, feed rates are optimized to based on the maximum chip thickness for each cut segment. The software takes into account where the material is in contact with the cutter and adjusts the feed rate to maintain a constant chip thickness. This is especially critical when tip cutting with a ball end mill, or when touring a surface with a small step-over. The feed rates continually change over the course of the cut in order to maintain a constant maximum chip thickness. The result is an improvement in tool life and a higher-quality surface finish.

Benefits of OptiPath

Faster time-to-market... Optimized tool paths reduce the over-all cycle time required to produce parts so they can be brought to market more quickly.

Save money... Reducing the time it takes to machine each part adds up to huge savings over the course of a year.

Increase productivity... Existing machine tools produce more parts in the same amount of time. It's almost like getting another NC machine for free!

Reduced fatigue... Machinists are not forced to keep a close eye on the machining process. They can set up other jobs or attend to other duties while the optimized program is running.

Reduced cutter wear... Less time required to cut each part means less cutter wear. Cutting at optimum conditions also prolongs tool life, so they are used longer before needing to be re-ground or replaced. Machinists are less frequently required to change tools or inserts.

Reduced machine wear... Machines are subject to less wear because optimized feed rates maintain a more constant cutting pressure between the machine tool and the workpiece. This reduces variable forces on the axis motors which leads to smoother NC machine tool operation.

Better quality... Optimized tool paths produce a better finish because constant cutting pressure causes little or no variation in cutter deflection. Finishes on corners, edges, and blend areas are significantly better. This reduces the amount of time required to bench finish the completed part.

High-speed machining—During high speed machining, the way the cutter contacts material is critical. Too low a feed rate produces chatter, vibration and work hardening, leading to poor surface finishes and premature cutter failure. Too high a chip load causes excessive cutting pressure and unsafe conditions that can lead to catastrophic failure of the tool, spindle, fixture and machine. Adjusting feed rates to maintain a constant chip load or volume removal rate help reduce these problems typically found in high speed machining – and are recommended by cutting tool manufacturers to solve the “chip thinning” problem. Constant chip thickness works especially well with high speed finishing with ball end mills where very small volumes of material are removed at high feed rates and spindle speeds. Spindle speed can also be optimized to maintain a constant surface speed at the maximum contact diameter on the cutter. Depending on the capabilities of the machine, constant surface speed optimization can help improve the surface finish.

Optimization settings—Feed rates are adjusted based on a number of user-defined settings. Feed rates are affected by machining conditions such as: machine tool capacities (horse-power, spindle type, rapid traverse speed, coolant, etc.); fixture and clamp rigidity; and cutting tool type (material, number of teeth, length, etc.).

Other settings are dependent on the nature of the tool path and determine the optimum feed rate for each cut:

Cut depth

Cut width

Cut angle

Entry feed rate

Cutter wear

Volume removal rate

The OptiPath Library—A library of feeds rates and surface speeds for different cutting tools, materials, and cutting conditions is included. Provided by Ingersoll Cutting Tool Company, this library serves as a useful starting point for determining the ideal feeds and speeds for particular machining conditions.

All optimization settings are stored in this central library, organized by type (different cutting tools, NC machines, etc.). Because everyone can access the same cutting information from the library, *OptiPath* is an excellent tool for achieving more consistent machining results between different operators, machines, shifts, etc. – even if your ‘resident expert’ is unavailable.

Acceleration / deceleration & NURBS support—*OptiPath* accounts for changes in velocity as each axis accelerates and decelerates during changes in direction. With this information, feed rates settings are based on the physical capabilities and limitations of the machine tool. *OptiPath* supports NURBS interpolation. It examines the machining conditions and amount of material removed by each segment of the curve – then calculates an ideal feed rate for each NURBS span, while maintaining the original geometric shape of the curve.

Model Export

With a few mouse clicks, the *Model Export* license enables you to export a cut model at any stage in the machining process, as an in-process CAD model complete with machined features!

CGTech’s unique algorithm minimizes file size, while preserving manufacturing features such as holes, fillets, corner radii, pocket floors and walls. The result is a highly accurate IGES or STL model.

Why Model Export?—Since the invention of the first NC machine tools, NC programming languages, and CAD/CAM systems, manufacturing engineers have needed an accurate electronic model of the initial and in-process state of material. It’s difficult to confidently create tool paths for each subsequent operation without an accurate representation of the initial material. The accuracy, efficiency, and “correctness” depend on the material’s initial geometric shape.

Access to the correct geometric workpiece shape in the CAD/CAM environment, at each manufacturing stage, helps avoid problems when creating subsequent cuts. Until now the only way to get an in-process CAD model is with expensive, labor-intensive, error-prone, and inaccurate methods. These include manually modeling the material state for each operation, or using scanning equipment and surfacing software on the physical workpiece produced by the previous operation.

Some CAD/CAM systems attempt to provide this by using their CAD solid modeling tools. This approach

Help on VERICUT

is typically limited to 2½ axis operations, and does not simulate motion. Instead it estimates the region to be removed, creates a solid feature for that region, and subtracts it from the initial stock model. Often details produced by cutter radius, step-over, or step-down are not included. And the size of the model can grow such that it becomes unmanageable.

Another CAD/CAM technique produces a rough faceted approximation of the in-process material. Typically limited to 3-axis motions and from a single setup position, the accuracy is only as good as the facet tolerance. Manufactured features such as holes, and radii are often lost. The models can be gigantic, requiring millions of facets to represent even relatively simple shapes.

Both of these techniques seek to avoid the difficulty of using an “as-machined” verification model to generate a usable CAD model. After simulating an NC tool path, the verification model contains much more data than a traditional CAD solid model algorithm can process. Because it is designed to create verification models, VERICUT has the unique advantage of being able to manage the data without failing and export an accurate CAD model in a manageable size.

Machine Simulation

In complex multi-axis machine work where tool orientation changes frequently, the possibility of costly collisions is a frightening reality. A machine crash can cost thousands of dollars, ruin the NC machine, and delay the entire production schedule. *Machine Simulation* detects collisions and near-misses between all machine tool components such as axis slides, heads, turrets, rotary tables, spindles, tool changers, fixtures, work pieces, cutting tools, and other user-defined objects.

VERICUT’s *Machine Simulation* enables you to perform realistic 3D simulation of entire NC machines, just like they behave in the shop, with extremely accurate collision-detection.

A complete library of pre-assembled machine tools is included. All models can be easily customized to match your machine. A partial list includes machines from: Bostomatic, Maho, Cincinnati Milacron, Makino, Mazak, Fadal, Mitsui Seiki, Okuma, SNK (Shin Nippon Kok), Kearney & Trecker, Sundstrand, Giddings & Lewis, Toyoda, Ingersoll Milling, and many more.

You can also build machine components and job-specific components (such as work pieces, cutting tools, rotary tables, spindles, fixtures, and clamps) in your CAD system and import them into VERICUT. By mixing imported CAD data with VERICUT solid primitives you can easily modify an existing machine or to create a new one. Any machines you modify or build from scratch can be saved to the machine tool library.

Simulating controls—VERICUT is developed by industry professionals who understand the complexities of NC code and control functions. This technical expertise makes VERICUT an excellent tool for verifying complex code using advanced programming functions such as:

- Look-ahead or 3D cutter compensation
- Tool tip programming & tool length compensation
- Gage length reference point programming
- Canned cycles & fixture offsets
- Rotary axis pivot points
- Variables, subprograms & macros
- Subroutines, looping & branching logic

With this license, you can accurately simulate every control in your shop and account for different

Help on VERICUT

machines, programs, parts, and functions. No programming language is required to simulate most NC controls. Machine Simulation includes an extensive library of flexible, easily modified controls, including: Fanuc, Allen-Bradley, Mazak, NUM, Cincinnati Acramatic, Okuma, Fadal, Maho Phillips, General Electric, Siemens Sinumerik, Giddings & Lewis, Heidenhain, Yasnac, and many more.

Collision checking—VERICUT is the fastest, most accurate NC machine collision checking program available.

Rather than checking only start and end points, or checking a number of points along a path, VERICUT uses a set of unique algorithms to check for potential collisions along the entire machine path.

The program also detects axis travel limit errors (user-defined limits for machine component movement). The program checks for and highlights collisions between all machine tool components such as axis slides, heads, turrets, rotary tables, spindles, tool changers, fixtures, work pieces, cutting tools, and other user-defined objects. The program also allows near-miss zones to be defined around the components.

Convert NC data—*Machine Simulation* is a powerful reverse post-processor. By translating NC programs to APT or other NC data formats, the program saves valuable machine tool and programmer time. Because obsolete or incompatible NC programs can be recycled or made to run on different machines, you can take advantage of all your programming resources – present and past.

Machine Developer's Kit

While VERICUT's capabilities are ideal for most, some users with special simulation or unusual processing requirements may require the Machine Developer's Kit – a high-level programming tool designed to increase VERICUT's ability to interpret and translate complex or unusual NC machine code data.

Do you need the Machine Developer's Kit?—VERICUT strives to support all the codes that are typically executed when running a job. However, it does not support all codes for every available control. (A special milling cycle, for example, may not be supported.) Under special circumstances some customization may be necessary. Based on a programming language similar to C, the *Machine Developer's Kit* provides a great deal of flexibility for interpreting and translating NC file formats. There are a number of additional benefits. For example, new functions can be written for new or enhanced control capabilities. Hardware and NC controls can be retrofitted or replaced without retiring NC code. VERICUT can be set to output NC code compatible with new equipment. Some programming experience is recommended.

AUTO-DIFF

AUTO-DIFF enables you to compare a design model to the “as-machined” VERICUT solid model to automatically detect differences and protect against gouges. Using this license, anyone involved in the production process can identify incorrectly machined areas. *AUTO-DIFF* is also used to detect a possible weakness or mistake in the design.

Why use AUTO-DIFF?—Often, from the time a part is designed to when it is ready to be machined, the design has passed through several people, departments, companies, and CAD/CAM systems. In the end, it is difficult to tell if the tool path accurately reflects the original design model. Without *AUTO-DIFF*, the only way to be sure is to cut the part and do a first article inspection. By using *AUTO-DIFF* to compare the simulated model to the design model, you are assured the tool path accurately represents the original design concept. This reduces the time it takes to cut the first production-worthy part. *AUTO-DIFF* also reduces the time it takes to prepare an NC tool path. You can check for gouges or excess material during the programming stage – not at the machine.

Solid, Surface, & Point comparisons

SOLID—Solid models from most popular CAD/CAM systems can be used. *AUTO-DIFF* embeds the solid model in the simulated part. You can identify gouges and excess material by setting a user-defined tolerance level. You can also compare the cut model with another VERICUT model – ideal for comparing different operations such as roughing and semi-finishing cuts.

SURFACE—A surface, a set of surfaces, or the skin of a solid model can be compared with the simulated part. The surface is overlaid on top of the simulated part, and gouges and excess material are displayed. Large amounts of data from very complex tool paths can be quickly compared. The design model does not need to be watertight, nor does the entire part need to be included.

POINT—When solid or surface model data does is not available, you can lay 3D points from CMM or manual methods over the top of the simulated part. The points are compared to the NC tool path to accurately detect errors or out-of-tolerance conditions.

Gouge checking—With *AUTO-DIFF*, you can perform constant gouge-checking interactively during simulation with virtually no degradation in speed or performance. It's not necessary to simulate the entire NC program first – you can identify and correct problems as they occur. To identify gouges, *AUTO-DIFF* embeds the design model inside the rough material. When the tool path cuts into the embedded model, the gouge is highlighted and the error is reported.

AUTO-DIFF also detects collisions between a tool and components nested inside the stock model, such as tooling pins, hardened inserts, or bearings. Previous cutting operations can be “protected” from any further machining. Any material removed from a protected area is highlighted and reported as an error. To make identifying gouges, collisions, or excess material easy, you can assign different colors to the design models, rough stock, and errors. Excess material boundaries can be exported as IGES or point data.

AUTO-DIFF reports—You can view automatically generates error reports which include information such as type of error, tolerance value, all tool path records out of tolerance, error location, and summary of results. *AUTO-DIFF* error reports are interactively linked to the graphics. Picking on an item in the report screen highlights the error on the screen. A report can be printed or saved as .TXT files. Both reports and graphical results can be generated in batch mode.

Catia Interface (CATV)

The CATV license provides a fully integrated CATIA-to-VERICUT interface that provides an improved way to work with the two programs. Using CATV, you can start and run VERICUT from inside CATIA. Once both programs are running, it's simple to toggle between the two.

CATV transfers CATIA geometry to VERICUT for stock, fixture, and design models. CATV also converts CATIA line and arc elements (two-dimensional tool profiles) into VERICUT cutter, shank, and holder tool definitions. Based on the CATIA coordinate systems, translations and rotations are automatically calculated.

CATV exports STL data from native CATIA models. With CATV there is no need for expensive STL converters. CATV outputs STL data directly from CATIA. And, unlike converters which only output STL data from solids, CATV outputs additional geometry such as faces, skins, surfaces, and volumes as STL data.

WorkNC Interface

This interface provides WorkNC users with an easy and convenient way to verify and optimize tool paths. VERICUT can be run directly from WorkNC, either interactively or in batch mode. To make setup easier, tool descriptions from WorkNC are automatically transferred to VERICUT. Stock blocks are created based

Help on VERICUT

on WorkNC settings.

UG Interface

The *UG Interface* license provides a fully integrated Unigraphics-to-VERICUT interface that automatically transfers all the UG data needed to simulate, verify and optimize your UG-generated NC tool paths.

Licenses for VERICUT for Pro/Engineer

The sections that follow describe the various configurations in which VERICUT for Pro/Engineer can be obtained from Parametric Technology Corporation. Each configuration is a complete verification system without prerequisites. All are floating licenses allowing VERICUT to run on any computer configured to run Pro/Engineer with VERICUT.

Licenses for VERICUT for Pro/Engineer:

Embedded Verification

NC Verification

NC Optimization

Machine Simulation

Note: VERICUT is also available for simulating NC data from other popular CAD/CAM systems, such as Unigraphics, CATIA, WorkNC, APT, etc. See Licenses for VERICUT by CGTech for more information.

Embedded Verification

CGTech's VERICUT tool path simulation, verification, and inspection system for NC machining is embedded in Pro/Engineer and accessible when a Pro/E Basic or Advanced NC license is available. With the Embedded Verification, you can simulate the material removal process and verify milling and drilling operations with 3-5 axes of simultaneous motion, as well as lathe turning operations. By interactively displaying the programmed tool path, VERICUT detects NC tool path errors, prevents NC machine tool crashes, and reduces or eliminates costly prove-outs. A batch processing capability is available to machine and inspect parts as well as record images of the simulation while you work in other applications or perform other duties.

The active Pro/Engineer workpiece, reference, and fixture models are automatically displayed in VERICUT with the CL-file tool path automatically oriented to the stock for proper NC machining. The cutting tools defined in the Pro/Engineer tool library are made available to VERICUT so that all you have to do to simulate the cutting process is press the "Play" icon (green arrow pointing right). Note: cutting tools that reference part models, known as "solid tools", are not currently supported in VERICUT, but may be supported in a future release.

Tools are available for inspecting and analyzing cut or uncut VERICUT models. Rotate, zoom-in on a part or portion of a part, display a part in translucent mode to inspect part features not normally visible in other applications, or cross-section a part for closer scrutiny.

The Pro/Engineer solid reference model is embedded in the stock workpiece. When machined, the steel blue color indicates all raw material is removed from the reference model. A red error color indicates the reference model is gouged, providing an opportunity to inspect the gouges before they occur on the actual cut part and contribute towards scrap loss. The AUTO-DIFF "Solid" difference method is available to compare the reference model against the simulated VERICUT part and reveal excess material left on the part.

NC Verification

Help on VERICUT

NC Verification upgrades the capabilities of Embedded Verification to increase the power of your verification system. Along with simulating the cutting action of the CL-file, you have the option to simulate the post-processed G-code data, and the ability to see and modify Pro/Engineer cutting tool shapes via VERICUT's Tool Manager. This upgrade also provides abilities to record images of the cutting action and verification process.

An available "FastMill" cut mode quickly processes very large 3-axis NC tool path programs. In this cut mode tool motions are not displayed as the system focuses on processing all the NC data to create the final cut part. When processing is complete, the display is updated with the resulting cut part. You have complete control of the speed, accuracy, and quality of the verification model and display.

NC Verification provides a robust cut model database with a complete history of all tool paths run on the workpiece. Interrogating cuts on the model can reveal the exact NC sequence and tool path record where errors will occur in machining. 3-D part features can be measured, including: stock thickness, volumes, depths, offsets, and more. Additional capabilities are provided for compare reference model data from Pro/Engineer or other sources against the simulated cut part. Solids, surfaces, and CMM inspection points can be used to highlight problem areas, such as gouge or excess material conditions.

Through recording images, *NC Verification* supports low cost, electronic document sharing between NC programmers and machine tool operators. By providing a preview of the machining process and a visual reference of the part before machining starts, production becomes more efficient.

NC Optimization

The *NC Optimization* upgrade includes all the capabilities of NC Verification, plus the ability to optimize NC tool paths and detect crashes with tool shanks or holders.

"OptiPath" NC feedrate optimization increases the speed and efficiency of the NC machining process. These increases produce substantial cost savings, enhanced quality of tooled parts, dramatic scrap reductions, and minimal wear and damage to NC machines. OptiPath reduces cycle times and boosts machining productivity. Optimization is controlled by user specified settings and VERICUT's constant knowledge of cutting conditions. Users can customize settings for a number of variable factors, even adjusting the tool path as the cutter dulls. By optimizing the tool path, parts are machined at a constant material removal rate. This enhances part quality by producing a better finish on surfaces and edges, along with minimizing scrap loss and tool wear.

NC Optimization adds flexibility and functionality to defining complex cutting tools. Users can define custom tool shapes beyond those available in Pro/Engineer to accurately depict the cutting tools used in NC manufacturing. Tool shanks and holders can also be described enabling VERICUT to check for collisions between these non-cutting portions of the tool and the workpiece or fixture.

Machine Simulation

The *Machine Simulation* upgrade provides the most comprehensive NC tool path simulation and verification system available by including all the capabilities of NC Optimization and more. With *Machine Simulation* you simulate post-processed G-code data to see how the tool will move on the actual machine tool. Complex NC control features are simulated, including: cutter diameter compensation, tool length compensation, work coordinate systems, gage length programming, programming with variables, NC macros, sub-programs, and looping or branching logic, and much more.

Machine Simulation comes with a library of pre-made NC machines and controls which can be used in any combination to interpret NC data destined for most popular mills, lathes, and wire EDM machining centers. A powerful user interface assists users with defining any number of additional NC machines and controls to create a realistic "digital manufacturing" environment capable of detecting almost any error in the NC

Help on VERICUT

manufacturing process.

3-D machines can be used to detect potential crashes with the workpiece setup or moving components, thereby avoiding potential scrap loss and costly machine down time. In addition to the traditional method of simulating tool path and machine motions (each in independent views), 2-4 axis lathe turning centers can be configured so material removal occurs on the machine. This realistic simulation technique provides collision checking with the 3-D turning tool assemblies and turret(s) against the rotating workpiece, holding fixture, and spindle.

Machine Simulation also provides the ability to convert ASCII NC data files into other NC data formats. By converting NC data formats, you save valuable machine tool and programmer time by making it possible to recycle obsolete or incompatible NC programs. Rather than reprogramming old tool paths, they can be converted and post-processed into a format compatible with current machine/control combinations. A log file is created showing the correlation between the NC data and records output from the conversion process.

Chapter 2 Using VERICUT

Overview of using VERICUT

VERICUT allows the user to simulate the NC machining process with software files, 3D models, and functions. Setting up a tool path to be processed by VERICUT is similar to setting up an NC tool path to run on an NC machine tool.

Requirements for a simulation

VERICUT requires three things to simulate the cutting action of a tool path, the same things that are required to machine the actual part on an NC machine tool:

1. **Stock**—raw material, or "workpiece" to be machined
2. **Tool path**—NC data describing cutting tool positions, machine information, and other data required to operate a machine tool
3. **Cutting tools**—shapes and sizes of cutting tools used to machine the workpiece

The items below are optional, however, can aid in the verification effort.

Fixture—hardware used to hold the workpiece for machining

Designed part—the theoretical perfect machined part

Operator instructions—notes about actions required by the machine operator which must be followed in order to successfully machine the part, such as: part setup, clamp/fixture changes, tool changes, etc.

The table below shows the requirements for setting up an NC tool path to run on an NC machine tool, and how these requirements are met in VERICUT.

Requirement	NC Machine Tool	VERICUT
Workpiece	Metal, or other stock material	Stock model
NC data- tool positions and NC machine codes	Tool path file(s)- G-code format	Tool path file(s)- APT-CLS or G-code format
Cutting tools	Cutting tools	Descriptions of cutting tool shape/size
Holding fixtures	Clamp/fixture hardware	Clamp/fixture models
Designed part	Not available until machined	CAD-designed model, or other design data
Operator instructions	Operator instructions	Operator instructions

VERICUT coordinate systems

VERICUT uses the coordinate systems below to: define and relate components and their models, define NC machines, and orient tool paths for proper cutting. Using **View menu > Axes**, you can display axes symbols representing these coordinate systems to see their relationships. Solid lines indicate axes are parallel or pointing out of the screen. Dashed lines indicate axes are pointing into the screen.

Component coordinate system—(XcYcZc) Coordinate system of a component. Each component has its own coordinate system. Components are defined and connected to other components via **Model menu** functions.

Model coordinate system—(XmYmZm) Coordinate system of a model. Each model has its own local coordinate system. Models are associated with components to provide 3-D properties. Models are defined and associated with components via **Model menu** functions.

Machine coordinate system—(XmcYmcZmc) Applicable only to machine views, this coordinate system is referenced when defining NC machines. X-caliper measurements in a machine view are relative to the machine coordinate system.

Workpiece coordinate system—(XwpYwpZwp) The coordinate system to which Stock, Fixture, and Design components are connected. This coordinate system only applies to workpiece views, and takes on a slightly different meaning, depending on if an NC machine is used in the simulation.

- *In a simulation where a machine is defined*, such as when processing G-code tool path files, the workpiece coordinate origin is the origin the machine component responsible for carrying the Stock, Fixture, and Design components.

- *In a simulation where a machine is NOT defined*, such as when processing APT-CLS tool path files, the workpiece coordinate origin is the origin of the non-moving "Base" component to which Stock, Fixture, and Design components are connected. In previous VERICUT versions, this was known as the "World coordinate system" (XwYwZw axes).

User-defined coordinate system—(XcsYcsZcs) A coordinate system defined by the user via the **Setup menu > Coord. System** function. The active coordinate system applies to X-caliper measurements, Section plane values, and tool path motions (except when orientations are defined in a tool path list).

A user-defined coordinate system is typically used to relocate an APT or CLS tool path for proper relationship to the workpiece, but can also instate different coordinate systems for defining section planes or gathering measurement data.

See the following topics in this chapter to learn more about using VERICUT:

The VERICUT main window

Introduction to VERICUT files

About VERICUT views

Color control in VERICUT

About modeling in VERICUT

Importing and exporting model and tool path data

Specify the tool path file(s) to simulate

Help on VERICUT

About describing cutting tools

Ways to monitor a VERICUT simulation

Image capture control in VERICUT

About building NC machines / About building NC controls

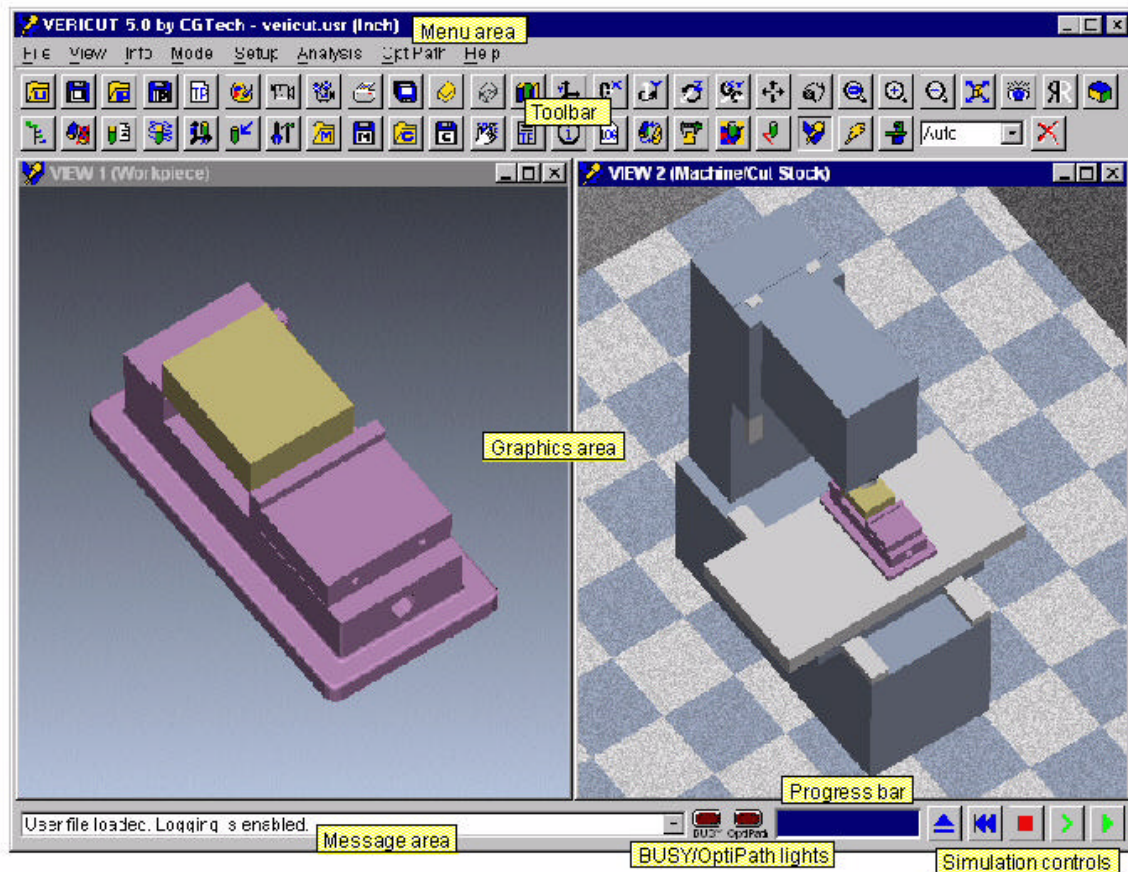
About optimizing tool paths

VERICUT windows

The VERICUT main window

The VERICUT main window is composed of distinct areas, each with different user interaction. The window header displays the last User file loaded and the current session units (inch or millimeter). This window can be resized like most other window, via dragging the window header, sides or corners.

To learn more about interacting with VERICUT through the main window components, click on the area you want to learn about.



Hint: Press "F1" in any VERICUT window to receive help on that window.

What do you want to do?

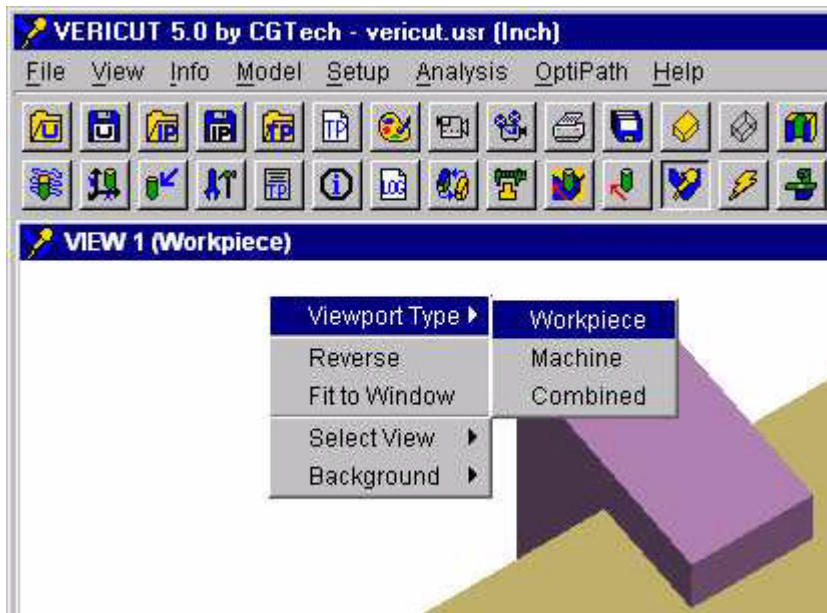
Change the look and feel of VERICUT's user interface, e.g. "Motif" vs. "Windows"

Learn about working with VERICUT files

Graphics area

The graphics area is where solid models of the workpiece, fixtures, etc. are displayed, and where the simulation takes place. By default, one view port is seen. Using **View menu > Layout** you can add more view ports displaying the workpiece or NC machine (if defined). All view ports are contained within the VERICUT main window.

Right-click in a view port to change its attributes, such as: view type, standard modeling views, background, etc. Use dynamic and static view options on the tool bar to rotate, zoom, or fit the model.



Many other VERICUT functions enable user interactions in this area, including: picking models or surfaces to be measured, determining tool path records responsible for specific cuts, and more. Consult the *Reference* chapter for more information.

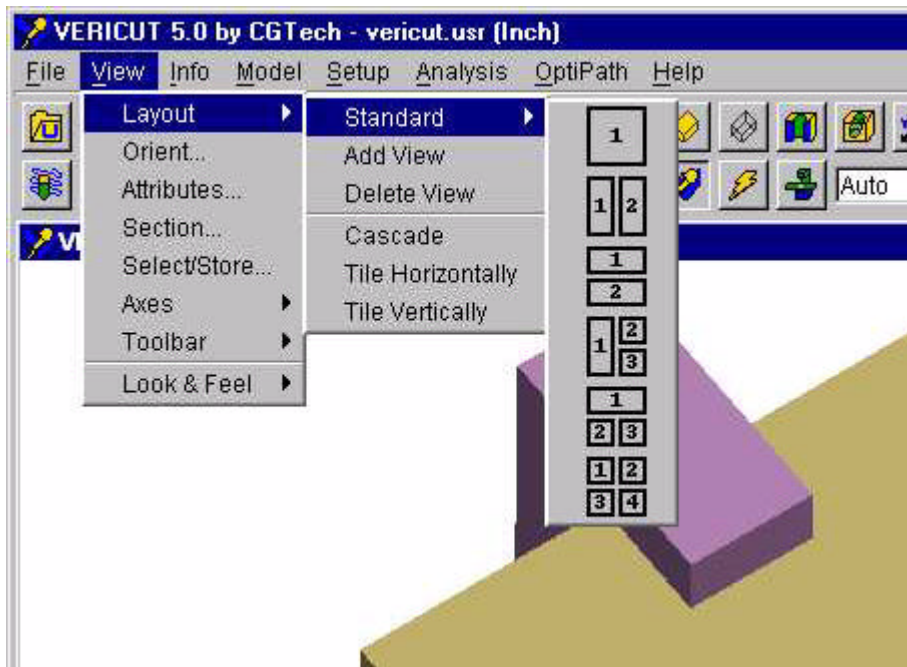
What do you want to do?

- Learn about other VERICUT main window components

- Learn about VERICUT viewing functions

Menu area

The menu located across the top of the VERICUT window provide easy access to VERICUT functions. Each menu contains groups of related functions. Left-click on any menu name to expose the list of functions in that menu. If an arrow appears to the right of a function in a menu, move the mouse over the arrow to expose a sub-menu of additional functions. Click on the function in the menu you want to use.



What do you want to do?

Learn about other VERICUT main window components

Learn about VERICUT functions

Toolbar

VERICUT's Toolbar provides quick and easy access to the most commonly-used functions/features.



To see what is associated with a Toolbar icon, simply position the cursor over the icon and a tip appears. You can close/open the Toolbar at any time. Using **View menu > Toolbar** you can also customize the Toolbar to suit your needs.

To close the Toolbar:

From the Toolbar select **Close Toolbar** .

To open the Toolbar:

1. Select **View menu > Toolbar**.
2. Select **Display Toolbar**.

To add/remove groups in the Toolbar:

1. Select **View menu > Toolbar**.
2. Clear check from a group to remove, put a check by a group to add.

Tip: You can affect all Toolbar groups at once using the **Set All / Clear All** buttons.

What do you want to do?

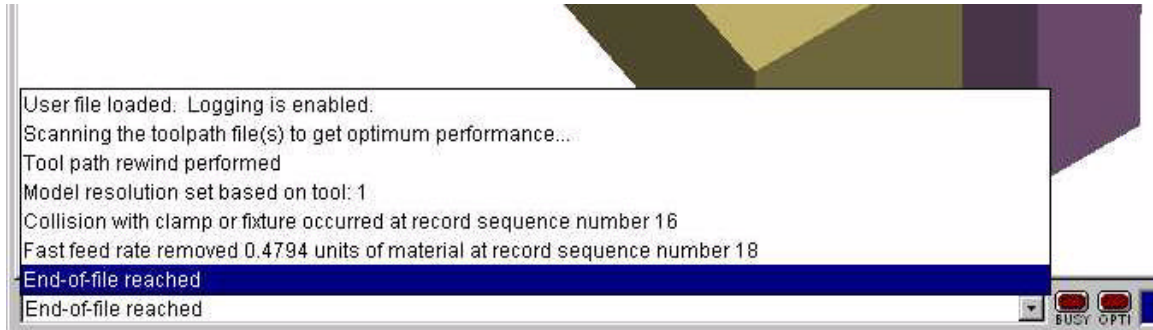
Learn about other VERICUT main window components

Learn what VERICUT functions the Toolbar can access

Message area

The message area, also known as the "message logger", is located below the graphics area, and by default displays the most recent message VERICUT has provided. VERICUT provides error, warning, and informational messages to assist you during a simulation session.

To review previous messages, click on the message line-the most recent message is listed last. Click again to close the message logger.



Note: All messages can be seen along with other valuable information in the Log file.

What do you want to do?

- Learn about other VERICUT main window components
- Learn about the Log file and other VERICUT files
- See problem descriptions and solutions for VERICUT error/warning messages

Progress bar

The progress bar located at the bottom-right corner of the VERICUT window provides visibility over progress of a CPU-intensive task, such as: comparing models using AUTO-DIFF, exporting model data, and checking the consistency of the stock model(s). The progress bar moves towards the right as the task nears completion. To interrupt the task, press the Stop icon (red square).

Sample progress bar nearing completion of a task:



What do you want to do?

- Learn about other VERICUT main window components

- Learn about VERICUT functions

Simulation (VCR) controls

The simulation controls, also known as VCR buttons, located at the bottom-right corner of the VERICUT window control interactive tool path simulation. Once VERICUT is configured, you will use these control to start and stop the simulation, as well as begin with a new, uncut workpiece. To see what is associated with a Simulation control icon, simply position the cursor over the icon and a tip appears.

Simulation control options:

Icon:	Name:	Function:
	Play to End	Start or re-start tool path processing
	Single Step	Process one tool path record ("single block")
	Stop	Stop processing (after the current record is processed)
	Rewind Toolpath	Rewind the tool path file to the beginning (leave the model as is)
	Reset Model	Display a new VERICUT model and rewind the tool path file

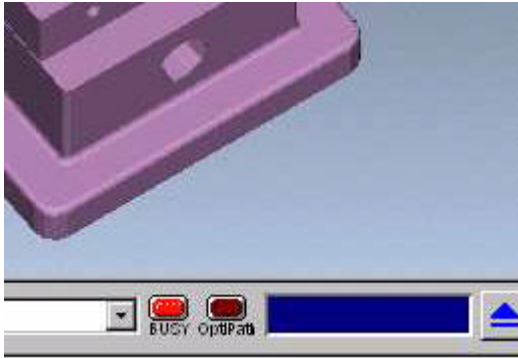
What do you want to do?

Learn about other VERICUT main window components

Learn about VERICUT functions

BUSY-OptiPath lights

The "BUSY" and "OptiPath" lights located at the bottom-right corner of the VERICUT window provide constant visibility over when VERICUT is busy performing a task, or optimizing a tool path file. A bright red color indicates an "on" condition, as seen in the example below with the "BUSY" light.



What do you want to do?

Learn about other VERICUT main window components

Learn about tool path optimization

Learn about VERICUT functions

Working with VERICUT files

Introduction to using VERICUT files

This section introduces you to files commonly used in VERICUT, and how to work with them. For a complete list of VERICUT files with detailed descriptions, see "File Types and Descriptions" in the *VERICUT Files* chapter.

Common VERICUT files

The files described below are most commonly used to configure VERICUT, monitor the verification session, and save session information. Familiarity with these files will enable you to work more efficiently in VERICUT, and get the most from your verification system.

Toolpath files—These files contain data describing cutting tool positions, machine information, and other information required to operate NC machine tools. VERICUT simulates and verifies two distinct types of NC tool path files: ASCII APT-CLS and G-code, or machine code data. VERICUT must be configured differently, depending on which tool path type you choose to process.

User files—These files contain VERICUT session settings, also known as "user configuration values" or "user values". Virtually everything changeable in the user interface can be saved in a User file. Relatively small in size, User files minimize effort required to configure VERICUT for processing a tool path file.

In Process files—Also known as "IP files", these files contain all data necessary to re-establish a VERICUT session, most importantly the 3-D VERICUT model with any cuts that were machined into it. Even the Log file (see below) is restored providing error history of prior tool path files used to machine the workpiece.

Log files and Results files— These files record information about the VERICUT session, such as: error, warning, and informational messages, machining times, optimization results, and more. Reviewing these files is an important step in ensuring that simulated tool paths have performed as expected.

Machine files and Control files— These files contain information required to simulate G-code tool path files and machining on an NC machine. Machine files contain such information as: machine axes, connect order, kinematics, 3-D size/shape, etc. Control files contain the rules for how machine codes are processed, which word/addresses perform specific functions, order of processing, etc.

Using the "Shortcut" feature

Using the **Shortcut** option list on file selection windows provides quick access to files in your Working Directory, or CGTech library and sample files. Choosing a shortcut automatically filters for files in that directory. You can also type new shortcuts into the list, but they are only remembered during the current VERICUT session.

Where VERICUT looks for referenced files

Help on VERICUT

VERICUT looks in the places listed below (in listed order) when trying to find files referenced in VERICUT files. An example follows.

1. Look in the directory specified by the path stored in the VERICUT file.
2. If not found, look in the current working directory.
3. If not found, look in the parent directory-directory where the file containing the referenced file came from.
4. If not found, look in the CGTech Library as specified by the CGTECH_LIBRARY environment variable.

Example- consider the following:

VERICUT is run from the **M:\mydirectory** location.

The command file used to run VERICUT defines CGTECH_LIBRARY to point to **C:\cgttech50\library**.

Using **File>Open**, you navigate to **D:\userfiles** and select a User file that contains a reference to a VERICUT Tool Library file stored in **E:\NCparts**.

VERICUT looks in the following locations to find the Tool Library file.

1. **E:\NCparts**—specified by the path stored in the VERICUT file
2. **M:\mydirectory**—current working directory
3. **D:\userfiles**—parent directory where the User file came from
4. **C:\cgttech50\library**—specified by the CGTECH_LIBRARY environment variable

VERICUT quits looking as soon as a file is found. If all locations are checked without finding the referenced file, an error is issued.

What do you want to do?

Set VERICUT to a "new" condition

Set a working directory

Save or open a VERICUT user configuration (User file)

Save or open a VERICUT model (IP file)

Save or open an NC control configuration (Control file)

Save or open an NC machine configuration (Machine file)

Create a G-Code or Control Report file

Edit the tool path file or a text file

View or reset the error Log file

Create and view a session Results file

Learn more about these and other VERICUT files

Setting VERICUT to a "new" condition

Use the procedure below to set VERICUT to a "new" condition. This action clears all session-related information as if no files were loaded. After performing this action, configure VERICUT by adding models, tool paths, descriptions of cutting tools, etc.

To set VERICUT to a "new" condition:

1. If needed, change the units for the new session as follows:
 - Click **File menu > Properties**.
 - On the **General** tab, select the desired **Default Units: Inch** or **Millimeter**.
 - Click **OK**.
2. Click **File menu > New**.

Tips:

The state in which VERICUT appears after performing a "new" operation is configurable. See "**File menu > New**" in the *Reference* chapter for details.

Setting a working directory

When a working directory is set, VERICUT will look here by default when files are saved or opened.

To set a working directory:

1. Click **File menu > Working Directory**.
2. In the window that opens, select or type the name of the directory.
3. Click **OK**.

The working directory is set. You can now quickly access files in this directory via the **Working Directory** entry in the file selection window **Shortcut** list.

Saving or opening a VERICUT user configuration

VERICUT session settings, also known as "user configuration values" or "user values", are savable in a User file. Virtually everything changeable in the user interface can be saved in a User file. Relatively small in size, User files minimize effort required to configure VERICUT for processing a tool path file.

To save a new User file:

1. Click **File menu > Save As**.
2. In the window that opens, select or type the name of the file to save.
3. Set **Units to Save** to the desired units in which to save file values.
4. Click **Save**.

The User file is saved. At a later time, you can open the User file as described below to reconfigure VERICUT, as described below.

To open an existing User file:

1. Click **File menu > Open**.
2. In the window that opens, select or type the name of the file to open.
3. If desired, set **Unit Conversion** to the desired unit measurement system.
4. Click **Open**.

If the VERICUT model was cut a message is displayed. Choose between the following:

- **Reset**—Erase the cut model, then load the incoming User file values and display a new model. Use this choice when the new User file contains a new model or view to be displayed.
- **Merge**—Merge the incoming User file with the current model. Use this choice when the new User file contains values for additional actions to be performed on the existing cut model.

The User file is opened. The current User file name and unit measurement system are displayed in the VERICUT window title bar.

Hint: Toolbar short cut for saving and opening User files: 

Saving or opening a VERICUT model

The 3-D VERICUT model with all machine cuts intact is savable in an In Process file, or "IP file". You can also save IP files to "backup" the VERICUT session, safeguarding against lost work if the session is interrupted by computer or power failure.

To save a new IP file:

1. Click **File menu** > **In Process** > **Save As**.
2. In the window that opens, select or type the name of the file to save.
3. Click **Save**.

The IP file is saved. You can open saved IP files as described below to reload the VERICUT model and continue working.

To automatically save IP files during processing:

1. Click **File menu** > **AutoSave**.
2. Click the **In Process** tab.
3. Under **Auto Save** and **Auto Error**, select the event(s) to output an IP file. If both are in use, be sure they have different **In Process File** names to avoid overwriting.
4. In the **In Process File** field, enter the base file name for saved files.
5. Press **OK** and continue cutting.

IP files are saved at the specified events.

To open an existing IP file:

1. Click **File menu** > **In Process** > **Open**.
2. In the window that opens, select or type the name of the file to open.
3. Click **Open**.
4. If the VERICUT model was cut, choose **Yes** to reset the model and load the IP file data.

The IP file is opened and the VERICUT model is restored.

Tips:

Toolbar short cut for saving and opening IP files: 

You can save IP files in batch mode using the **File menu** > **AutoSave** configuration described above.

Saving or opening an NC machine configuration

The NC machine configuration, including machine construction, kinematics, and other properties, is savable in a Machine file. When combined with a Control file, VERICUT can simulate how an NC machining center will respond to G-code tool path files.

Use the procedures below to save and open Machine files. Note that many machine configurations are available in the CGTech library- see "Machine files" in the *VERICUT Files* chapter for more information.

To save a new Machine file:

1. Click **Setup menu > Machine > Save As**.
2. In the window that opens, select or type the name of the file to save.
3. Set **Units to Save** to the desired units in which to save file values.
4. Click **Save**.

The Machine file is saved. You can open saved Machine files as described below to reload the NC machine configuration.

To open an existing Machine file:

1. Click **Setup menu > Machine > Open**.
2. In the window that opens, select or type the name of the file to open. (**Hint:** Use the CGTECH_LIBRARY shortcut to see machines in the CGTech library.)
3. Click **Open**.

The Machine file is opened and all machine views are updated. Machines that do not have 3-D models will not appear. **Note:** Always reset VERICUT (press ) after opening or changing Machine or Control files.

Tips:

Toolbar short cut for saving and opening Machine files:  

For details about CGTech library machines and controls, and other sample files you can reference when simulating G-code tool path files, see the *VERICUT Files* chapter.

Saving or opening an NC control configuration

The NC control configuration, including rules for how machine codes are interpreted, associated subroutines, and more, is savable in a Control file. When combined with a Machine file, VERICUT can simulate how an NC machining center will respond to G-code tool path files.

Use the procedures below to save and open Control files. Note that many control configurations are available in the CGTech library- see "Control files" in the *VERICUT Files* chapter for more information.

To save a new Control file:

1. Click **Setup menu > Control > Save As**.
2. In the window that opens, select or type the name of the file to save.
3. Set **Units to Save** to the desired units in which to save file values.
4. Click **Save**.

The Control file is saved. You can open saved Control files as described below to reload the NC control configuration.

To open an existing Control file:

1. Click **Setup menu > Control > Open**.
2. In the window that opens, select or type the name of the file to open. (**Hint:** Use the CGTECH_LIBRARY shortcut to see controls in the CGTech library.)
3. Click **Open**.

The Control file is opened **Note:** Always reset VERICUT (press ) after opening or changing Machine or Control files.

Tips:

Toolbar short cut for saving and opening Control files:  

For details about CGTech library machines and controls, and other sample files you can reference when simulating G-code tool path files, see the *VERICUT Files* chapter.

Creating G-Code or Control Report files

A G-Code Report file contains information about how the current NC control configuration will interpret a G-code tool path file. Reported information includes: machine codes present in the tool path and macros (actions) they will call, variables used (if any), cutting tools used, subroutines defined and referenced, and more. A Control Report file is similar to the G-Code report, but contains more detailed information about all machine codes and features supported by the NC control.

To generate a report of G-codes in a tool path file and what the control will do with them:

1. Assuming the desired G-code tool path and Control files are loaded, click **Info menu > G-Code Report**.
2. Select or enter the report file name, and click **OK**.

The report is generated and displayed for review.

To generate a report of all machine codes a control is configured to handle:

1. Assuming the desired Control file is loaded, click **Info menu > Control Report**.
2. Select or enter the report file name, and click **OK**.

The report is generated and displayed for review.

Editing the tool path file

VERICUT provides a fast and easy way to edit your tool path file. The procedure below assumes you have already specified the tool path file for VERICUT to process. (Ref. **Setup menu > Toolpath**)

To edit the current tool path file:

1. Click  on the toolbar, or click **File menu > Edit > Toolpath File**.
2. Edit the file as desired.
3. In the editor window, click **File menu > Save**, then close the editor window.

Warning: Editing the tool path file can cause undesirable results, especially when edits are made to portions of the tool path which have already been processed by VERICUT. These edits can cause the tool path record number pointer to be out of sync with the edited tool path file. In these cases it is recommended to rewind the tool path file after editing, and restart the simulation with the last unedited record prior to the first addition/deletion.

Editing an ASCII text file

VERICUT provides a fast and easy way to edit any ASCII text file. All standard text editing features are supported, such as: copy/cut, paste, search, etc.

To edit an ASCII text file:

1. Click **File menu > Edit > Text File**.
2. Edit the file as desired.
3. In the editor window, click **File menu > Save**, then close the editor window.

View the tool path file while processing

Use the procedure below to open a window containing the tool path file being simulated. Scroll through the entire file, search for text forward and backward, print, and more. An arrow pointer indicates the current tool path record being processed.

To see the current tool path file:

Click  on the toolbar, or click **Info menu > Toolpath**.

View the status of simulation and machine conditions

Use the procedure below to see status of errors and warnings detected by VERICUT, machine and tool tip locations, information about the cutting tool, machining conditions, and more. You can close this window, and reopen it at any time to view current status.

To view simulation status:

1. Click  on the toolbar, or click **Info menu** > **File Status**.
2. If the desired information is not shown, press **Configure** to reconfigure window display.

Viewing and resetting the error Log file

The VERICUT Log file contains session information, such as error, warning and informational messages about the verification session.

To view the Log file:

1. Click  on the Toolbar, or click **Info menu > VERICUT Log**.

A VERICUT Log window opens containing the Log file.

To reset (clear) Log file contents:

1. Click  on the Toolbar, or click **Info menu > VERICUT Log**.
2. In the VERICUT Log window, click **File menu > Reset Log File**.
3. Choose **Yes** to clear the Log file contents.

Tips:

Be sure to check for errors and warnings in the Log file. If present, changes to the tool path file may be required.

For help with interpreting error and warning messages in the Log file, consult the *ERRORs and Problem Solving* chapter.

Creating and viewing a session Results file

The Results file is an HTML file that contains overview information about the VERICUT session, such as: the current tool path and User files, part images, machining times, optimization results, and links to the error Log file and optimized tool path file (if any). Use the procedures below to create and view this file.

To create a Results file:

1. Ensure VERICUT is configured to create the desired data for the Results file, for example: auto-saving images, optimizing, etc.
2. If desired, specify the name of the Results file-use the field on the **File menu > Properties, Results** tab to do this.
3. Complete your verification/optimization session.

The Results file is created automatically by VERICUT, and can be viewed as described below.

To view the Results file:

Click **Info menu > Results File**.

The Results file is opened in your HTML browser. There are many things you can do from this file-see "Results file" in the *VERICUT Files* chapter for more information.

Working with views

About VERICUT views

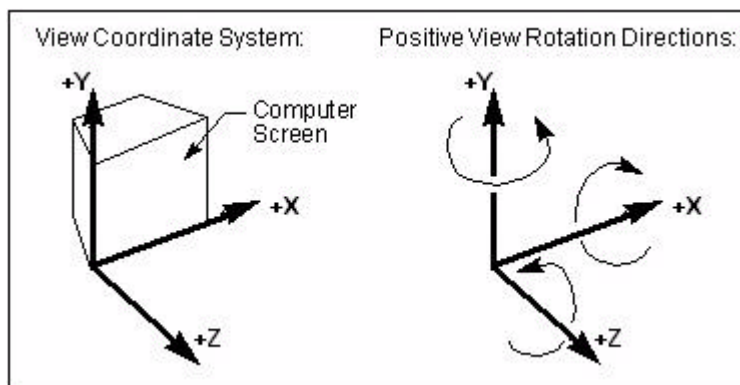
One of the most powerful advantages of VERICUT's user interface is flexibility in viewing the simulation. The simulation can be viewed from any distance or direction, in as many views as required to gain good visibility of the machining process. Just a click away, you can change the quantity and layout of view ports, each capable of showing machining on the workpiece, the NC machine, or a "combined" view where machining takes place on the machine.

Orienting views

Objects can be quickly oriented in standard drawing views, dynamically manipulated, and sectioned to "virtually see" every aspect of machining. VERICUT provides tools to rotate, zoom, pan, and reverse models in any view.

The "View coordinate system"

View rotations are performed about X, Y, or Z axes of a right-hand Cartesian "view coordinate system". The view coordinate system origin is oriented such that the left edge of the computer screen is the Y-axis, the bottom edge is the X-axis, and the Z-axis is pointing out of the screen towards you.



View attributes

View attributes add to visualization power by enabling you to:

- Change what is seen in a view: workpiece machining, machine motions without machining, or machining a workpiece on a machine

- Show machines in shaded or wireframe display

- Control light direction

- Change the background, including ability to add walls and a floor in a view of an NC machine.

Each view has its own individually controlled attributes.

What you can do with views

The sections that follow describe how to work with views to get the most out of the simulation. See the appropriate section for steps to accomplish the listed task.

What do you want to do?

View-related:

- Change the view layout

- Add or delete a view

- Cascade or tile views

- Change the view background

Model-related:

- Rotate a view of the model

- Reverse a view of the model

- Zoom a view of the model

- Pan a view of the model

- Fit models in a view

- Section a model

- Change how a machine is drawn

View attribute-related:

- Change the view type

- Change how a machine is drawn

- Change the light source direction

- Change the view background

Custom/stored views:

- Store and select custom views

- Modify or rename a stored view

- Delete a stored view from the view list

Rotating a view

You can rotate views of the workpiece, NC machine, or cutting tools seen in the Tool Manager to provide optimum viewing. View rotations are described with respect to the "view coordinate system"- see "About VERICUT views" in the *Using VERICUT* chapter for details. VERICUT offers several methods of view rotation, as described below.

View orientation does not change the orientation of the model in physical space. To do this, use the **Model menu > Model Definition** function instead.

To dynamically rotate a view:

1. Click in the view to rotate.
2. Press and hold the <Ctrl> + <Shift> keys while left-clicking and dragging the mouse in the direction to rotate.

The model rotates in the direction you drag. Note that the speed of rotation varies with the complexity of the model.

You can also rotate specifically about view coordinate system X, Y, or Z axes using the dynamic

rotation options 

on the Toolbar or in the **View menu > Orient** window.

To rotate a view by angle values (degrees):

1. Click in the view to rotate.
2. Click **View menu > Orient**.
3. *To enter view coordinate system X Y Z angles:* enter the values for X rotation, Y rotation, Z rotation (separated by spaces) in the **Angles** field and press **Apply**.

For example, "0 0 0" specifies no rotation, or XY "plan" view. "-30 0 0" rotates 30 degrees in the negative direction about the view coordinate system X-axis.

4. *To rotate incrementally by degrees:* enter the degrees to rotate incrementally in the **Increment** field, then press the desired rotation axis/direction button: **X+** **X-**, ... etc.

After a short delay, the view of the model is rotated.

To rotate a model 180 degrees (reverse) in a view:

1. Click in the view to rotate.
2. On the Toolbar or in the **View menu > Orient** window, click **Reverse** .

After a short delay, the view of the model is reversed.

Tips:

If object display appears coarse, use **Refine Display**  to improve image quality.

If you create a custom view and expect to use it frequently, use **View menu > Select/Store** to

Help on VERICUT

store the view in a list where it can be quickly retrieved at any time.

While manipulating views it may be helpful to display coordinate systems associated with models and tool paths. To do this, click **View menu > Axes** and select the axes you wish to display.

Reversing a view

You can reverse views of the workpiece, NC machine, or cutting tools seen in the Tool Manager at any time. The affect is as if you walked behind the computer screen and viewed the model from the opposite side.

Reversing the view does not change the orientation of the model in physical space. To do this, use the **Model menu > Model Definition** function instead.

To reverse a view:

1. Click in the view to reverse.
2. Click  on the Toolbar. or click **View menu > Orient** window, then click **Reverse**.

After a short delay, the view of the model is reversed.

Tips:

If object display appears coarse, use **Refine Display**  to improve image quality.

While manipulating views it may be helpful to display coordinate systems associated with models and tool paths. To do this, click **View menu > Axes** and select the axes you wish to display.

Zooming a view

You can magnify (zoom in) or reduce (zoom out) the view of a workpiece, NC machine, or cutting tools seen in the Tool Manager. VERICUT offers several methods of zooming, as described below.

To dynamically zoom:

1. Click in the view to zoom.
2. Press and hold the <Ctrl> key while left-clicking and dragging the mouse mouse up to zoom in (magnify), down to zoom out (reduce).

You can also zoom using the dynamic zoom option  on the Toolbar or in the **View menu > Orient** window.

To zoom by "rubberbanding" a target area (static zoom):

1. Click in the view to zoom.
2. Click  on the Toolbar, or click **View menu > Orient** window, then click **Zoom to Box**.
3. Click-left to anchor a corner of the zoom box.
4. Move the mouse to stretch the box around the area to zoom.
5. When the target area is surrounded, click again to indicate the opposite corner of the box.

To cancel the operation while rubberbanding a zoom box press the <Escape> key.

To zoom in or out by steps (approx. 20% each step):

1. Click in the view to zoom.
2. Click the desired option on the Toolbar:  , or click **View menu > Orient** window, then click **Zoom In/Out**.

Tips:

If object display appears coarse, use **Refine Display**  to improve image quality.

If you create a custom view and expect to use it frequently, use **View menu > Select/Store** to store the view in a list where it can be quickly retrieved at any time.

While manipulating views it may be helpful to display coordinate systems associated with models and tool paths. To do this, click **View menu > Axes** and select the axes you wish to display.

Panning a view

You can pan (shift) views to see different portions of the workpiece, NC machine, or cutting tools seen in the Tool Manager. VERICUT offers several methods of panning, as described below.

To dynamically pan:

1. Click in the view to pan.
2. Press and hold the <Shift> key while left-clicking and dragging the mouse mouse in the direction to pan.

You can also pan using the dynamic pan option  on the Toolbar or in the **View menu > Orient** window.

To pan by picking "from" and "to" positions (static pan):

1. Click in the view to pan.
2. Click **Pan**.
3. Click-left to indicate the move "from" point.
4. Move the mouse and click again to indicate the move "to" point.

To cancel the operation while positioning for the move "to" point, press the <Escape> key.

Tips:

If object display appears coarse, use **Refine Display**  to improve image quality.

If you create a custom view and expect to use it frequently, use **View menu > Select/Store** to store the view in a list where it can be quickly retrieved at any time.

While manipulating views it may be helpful to display coordinate systems associated with models and tool paths. To do this, click **View menu > Axes** and select the axes you wish to display.

Fitting models in a view

A "Fit" operation resizes the workpiece, NC machine, or cutting tools seen in the Tool Manager so the entire model can be seen in the view.

To fit a model in a view:

1. Click in the view to fit.
2. *To fit a single view:* on the Toolbar or in the **View menu > Orient** window, click **Fit** .
3. *To fit all views:* click **View menu > Orient**, then click **Fit All**.

Tips:

If object display appears coarse, use **Refine Display**  to improve image quality.

While manipulating views it may be helpful to display coordinate systems associated with models and tool paths. To do this, click **View menu > Axes** and select the axes you wish to display.

Sectioning a model

Use the procedures below to section a VERICUT model, then restore the original solid. Sectioning removes the forward visual portion of the model. Sectioning only applies to the active view. However, models in any view can be sectioned.

To section a VERICUT model:

1. Click  on the toolbar, or click **View menu > Section**.
2. Under **Plane 1** for example, choose the **Plane Type** which represents the tool path axis to section perpendicular to, or choose "**Custom**" to define a plane using **Distance** and **Normal I J K** values.
3. *To section at an exact plane location-* enter the **Distance** (measured along the normal vector) to establish the section plane, then press **Apply** or **OK**.
4. *To section through a point on the model selected by the mouse-* select **Section on mouse pick**, then click on the model at the location to section.

The model is sectioned. You can continue cutting in the sectioned view, or restore the original solid as described below.

To restore the original solid from a sectioned view:

1. Click  on the toolbar, or click **View menu > Section**.
2. Click **Restore**.

If in "mouse pick mode", you can also click in the graphics background (off the model) to restore the model.

To display an animated series of automatic section cuts:

1. Click  on the toolbar, or click **View menu > Section**.
2. Ensure that the **Section on mouse pick** checkbox is cleared (no check).
3. In the automatic sectioning portion of the window, choose the plane which will set the orientation for automatic sectioning. If necessary, choose the **Plane Type** for the selected plane.
4. To set the starting section plane, click in the **Start** field, then click on the model (or type) to supply a start-distance value.
5. To set the ending section plane, click or type to supply an **End** field value.
6. Enter the number of **Cuts** to perform, and **Delay** (in seconds) to pause between each cut.
7. Click **Auto**.

The section animation begins. When automatic sectioning is finished, you can continue cutting in the last sectioned view, or restore the original solid as described above.

Changing the view layout

Use the procedure below to change the view layout in VERICUT.

To change the view layout:

1. Click **View menu > Layout > Standard**.
2. Choose the option that provides the desired view quantity and layout.

The view layout set. More views can be added, or views can be deleted from the current layout using other functions in **View menu > Layout**.

3. Drag or resize any of the view ports in the VERICUT main window as desired via dragging the view port window header, sides or corners.

Adding or deleting views

Use the procedure below to add or delete view from the VERICUT view layout.

To add a new view:

From the **View menu > Layout > Add View**.

A new view is added. Drag or resize views via dragging the view window header, sides, or corners.

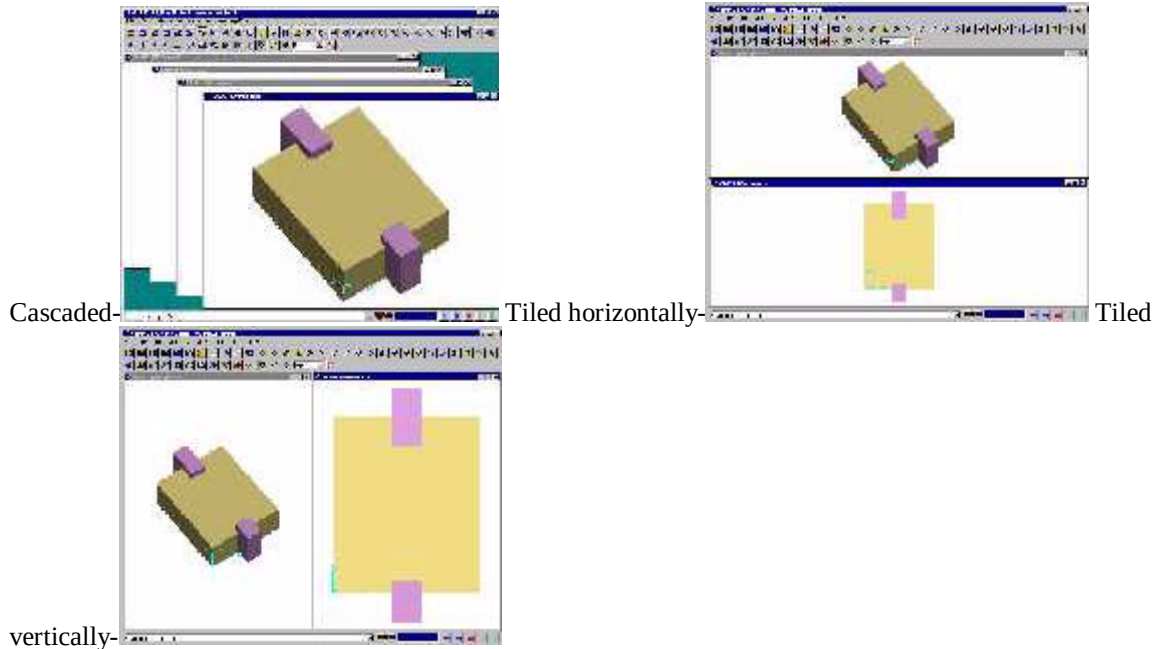
To delete a view:

1. Click in the view to delete.
2. From the **View menu > Layout > Delete View**.

Shortcut: You can also delete a view via clicking "X" at the top right corner of the view window.

Cascading or tiling views

Use the procedure below to arrange views in a cascaded or tiled layout. Example arrangements follow.



After any of these operations, you can drag or resize any of the views via dragging the view window header, sides or corners.

To cascade views:

From the **View menu > Layout > Cascade**.

The view ports are arranged in a cascaded (overlapping) arrangement.

To tile views horizontally or vertically:

From the **View menu**, select **Layout > Tile Horizontally** or **Vertically**, depending on preference.

Changing the view type

VERICUT enables you to see machining in a variety of ways. Each view is capable of displaying the workpiece being machined, the NC machine, or a "combined" view of the workpiece being cut on the NC machine.

To change the view type to display a workpiece, machine, or combined view:

1. Right-click in the view to change.
2. From the displayed menu, select the desired **View Type**.

Note: A 3-D machine must be defined to see it in a **Machine** or **Machine/Cut Stock** view.

Tips:

You can change the **View Type** and access other view attributes via the **View menu > Attributes** function.

Changing the draw mode of a machine

VERICUT can draw a 3-D machine in many ways: shaded solids, wireframe, hidden lines, etc. How the machine is drawn has no affect on the accuracy of the simulation or collision checking. However, can affect performance. The affect varies, depending on your hardware configuration, and complexity of the machine/workpiece.

To change how the machine is drawn:

1. Click in the view to change the machine.
2. Click **View menu > Attributes**.
3. Choose the desired **Draw Mode** for the machine:
 - **Shade**—shaded solids
 - **Lines**—wireframe
 - **Hidden**—wireframe with back-facing lines hidden
 - **Mixed**—shade, lines, or hidden-as defined by each component
4. Press **Apply** or **OK**.

Tips:

Toolbar short cut for shaded and wireframe draw modes:



You can change the **Draw Mode** and access other view attributes via the **View menu > Attributes** function.

Changing the direction of the light source

When model features can not be seen well due to poor lighting, there are two options:

1. Change the light brightness (via **File menu** > **Colors: Brightness**), or
2. Change the light source direction

Light source I J K vector values are described with respect to the "view coordinate system"- see "About VERICUT views" in the *Using VERICUT* chapter for details. Changing the light source direction has no affect on simulation speed or accuracy.

To change the light source direction:

1. Click in the view to change the light source.
2. Click **View menu** > **Attributes**.
3. In the **Light** field, enter I J K vector values (separated by spaces) which define the view coordinate system vector direction in which to point the light.
4. Press **Apply** or **OK**.

The light source is updated and model shaded accordingly. A second source is automatically defined to light the opposite side of the model.

Change the view background

VERICUT offers several background types for views. The choice of background has no affect on simulation speed or accuracy, but can make the simulation appear more attractive or realistic.

To change the background in a view:

1. Right-click in the view to change backgrounds.
2. From the displayed menu, select the desired **Background**.

Tips:

For walls to display correctly, ensure that **Machine Up** is set for your machine. (Ref. **Setup menu** > **Machine** > **Settings**)

Colors for backgrounds are defined via the **File menu** > **Colors** function.

You can change the **Background** and access other view attributes via the **View menu** > **Attributes** function.

Storing and selecting your own views

Defining custom views provides quick ways to see all important aspects of machining. A custom view can be named, then stored and selected from a "View List". There is no limit to the number of views you can store.

To store a custom view:

1. Orient the model to create the view to store.
2. With the custom view active, click **View menu > Select/Store**.
3. Press **Add**.
4. In the window that opens, enter a unique **View Name**.
5. Press **OK**.

VERICUT adds the view after the selected view in the list. You can use a stored view at any time, as described below.

To select a stored view:

1. Right-click in the view to change.
2. From the displayed menu **Select View** list, select the view to use.

VERICUT displays the model in the selected view.

Modifying or renaming stored views

Use the procedure below to update custom views stored in the View List.

To modify a stored view:

1. Open the **View menu > Select/Store** window and select the view to modify from the View List.
2. Change the view orientation as desired, for example: rotate, zoom, etc.
3. Press **Modify**, then **OK** to confirm the modification.

Note that the **View Name** is shown for reference purposes only, and can not be edited in this window. Use the procedure below to edit the view name.

To rename a stored view:

1. Click in a view to make it active for changes, then click **View menu > Select/Store** .
2. Choose the view in the View List to rename.
3. Enter the new **View Name**.
4. Press **Rename**, then **OK** to confirm the new name.

Deleting stored views from the view list

Use the procedure below to delete stored views from the View List.

To delete a stored view:

1. Open the **View menu** > **Select/Store** window and select the view to delete from the View List.
2. Press **Delete**.
3. Press **Delete** to delete the selected view, or **Delete All** to delete all defined views.

Using colors

Color control in VERICUT

VERICUT uses colors to distinguish between different types of models, machine components, and cutting tools. While the default color scheme in VERICUT is adequate for most users, you can manipulate the colors you see in a variety of ways.

The sections that follow describe various ways you can use colors in VERICUT.

What do you want to do?

- Add or delete display colors
- Assign a color to the cutting tool
- Assign colors to cut sequences
- Change the cut color during a cut
- Reset the cut color of all cuts
- Change background or foreground colors
- Change light brightness
- Change the "error color"

Adding or deleting display colors

Use the procedure below to add or delete display colors. The quantity of display colors can be changed at any time (maximum 12 colors).

To add (or delete) colors that can be displayed by VERICUT:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Define** tab.
3. *Delete a color*- select the color in the list to delete, then click **Delete**.
4. *Add a color*- click **Add**.
5. *Change a color*- slide the **Red, Green, Blue slide bars**, or click **Color List** to choose from pre-mixed colors.
6. If desired, edit the color name by clicking in the corresponding text field, then editing.
7. Press **OK** or **Apply**.

The new colors are applied to your session. You can assign the colors to solid objects using Assign tab features.

Assigning a color to the tool

By default, VERICUT displays cutting tools in colors associated by the **Cut Seq. / Color list**. If preferred, you can shade all cutting tools displayed by VERICUT in a specified color. Doing so does not affect cut color-color of surfaces machined by the tool. See "Assign colors to cut sequences" for control over cut colors.

To assign a color for all cutting tools:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Assign** tab.
3. Choose the desired **Tool** color.

If the color you want is not available, add a new display color as described by "Add or delete display colors" in this chapter.

4. Press **OK** or **Apply**.

Assigning colors to cut sequences

"Cut colors" are the colors used on material removed by a tool using a safe cutting feed rate. By default, the **Cut Seq. / Color list** under the **File menu > Colors** function determines the cut colors, and order in which they are used.

To add (or delete) cut colors:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Assign** tab.
3. *Delete a cut color-* select the color in the list to delete, then click **Delete**.
4. *Add a cut color-* click **Add**.
5. Choose the desired color for the cut sequence.

If the color you want is not available, add a new display color as described by "Add or delete display colors" in this chapter.

6. Press **OK** or **Apply**.

Change the cut color during a cut

Use the procedure below to assign a new cut color during a cut sequence. You can make this change without affecting previous cuts made by the tool, or repaint all of the tool's cuts.

To change or repaint a cut color while cutting:

1. Click  on the Toolbar, or click **File menu > Colors**.
2. Click the **Assign** tab.
3. In the **Cut Seq. / Color list**, select the cut color to change.
4. Choose the desired color for the cut sequence.

If the color you want is not available, add a new display color as described by "Add or delete display colors" in this chapter.

5. *To keep the color of previous as is, and only affect future cuts-* press **OK** or **Apply**.
6. *To re-color all of the tool's cuts-* press **Repaint**.

Resetting the color of all cuts in a workpiece

Resetting the cut color can be useful to distinguish new cuts on a model that has already been cut, such as between roughing and finishing operations. Previously cut areas (areas cut before resetting the cut color) can still be differentiated from true uncut stock by applying a new **Reset Cut** color.

To reset the color of all cuts in a workpiece:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Assign** tab.
3. If desired, choose a **Reset Cut** color.

By default, the **Reset Cut** color is set the same as the stock model to give the appearance of uncut stock, however, any defined color can be selected. If the color you want is not available, add a new display color as described by "Add or delete display colors" in this chapter.

4. Press **Reset Cut Color**.

Changing the background or foreground colors

Use the procedure below to change the color of the graphics area background, or color of items drawn in the foreground, such as: coordinate system axes, selected model/component boundary lines, etc.

To change the background or foreground colors:

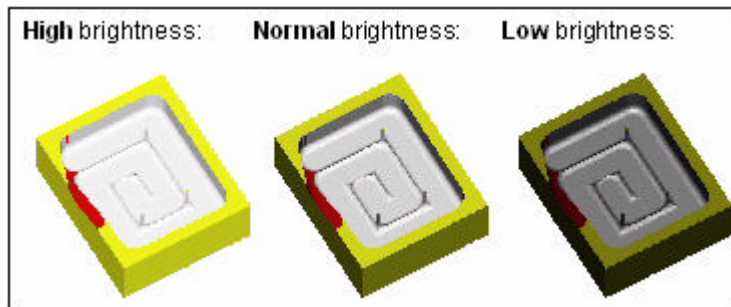
1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Define** tab.
3. Select **Background** (or **Foreground**).
4. Change the color via sliding the **Red, Green, Blue slide bars**, or click **Color List** to choose from pre-mixed colors.
5. If desired, edit the color name.
6. Press **OK** or **Apply**.

Changing the light brightness

Use the procedure below to change how bright the lighting is for shading solid objects, such as: models, the tool, NC machine components, etc.

To change the amount of brightness for lighting:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Define** tab.
3. Choose the desired level of **Brightness**.



4. Press **OK** or **Apply**.

Tips:

High brightness shows more detail in poorly lit areas of the model.

Low brightness can make small deviations in the model surface more obvious, such as gouges in molds/dies, etc..

Changing the "error color"

By default, a vibrant red color is used to highlight errors and unsafe machining conditions, such as:

Material removed while cutting too fast (known as a "fast feed" error)

Cuts in clamps, fixtures, bolts, and other hardware holding the workpiece

Material removed by non-cutting portions of the tool, such as: tool holders, extensions, etc.

Gouges in a design model (via AUTO-DIFF)

If necessary, you can change the error color to make it easier for you to see.

To change the color for highlighting error conditions:

1. Click  on the toolbar, or click **File menu > Colors**.
2. Click the **Assign** tab.
3. Choose the desired **Error** color.

If the color you want is not available, add a new display color as described by "Add or delete display colors" in this chapter.

4. Press **OK** or **Apply**.

When you see the error color on a VERICUT model, use **History** in X-Caliper, or see the tool path history information associated with the error in the Log file to identify the responsible tool path record.

Modeling in VERICUT

About modeling in VERICUT

VERICUT provides a number of ways to describe the shapes and attributes of solid objects used in a simulation. You can import IGES, STL and VERICUT model files, or use VERICUT's modeling capabilities to define simple block, cylinder, or cone-shaped models.

VERICUT uses components to describe how to use various solid objects. 3-D size and shape is provided via adding one or more models to components. Each of these concepts are described in more detail by the sections that follow.

Components

A component defines the function of a solid object. VERICUT provides many component types to describe the solid objects used in a simulation. A few easily recognized examples are: Stock, Fixture, Design, Tool, and various machine axes (XYZ, ABC, UVW, etc.). VERICUT determines how to handle solid objects based on the component's type and attributes. For example, VERICUT assumes a Stock-type component represents a workpiece to machine, while a Fixture-type component is intended to hold the workpiece for machining (and is not expected to be machined).

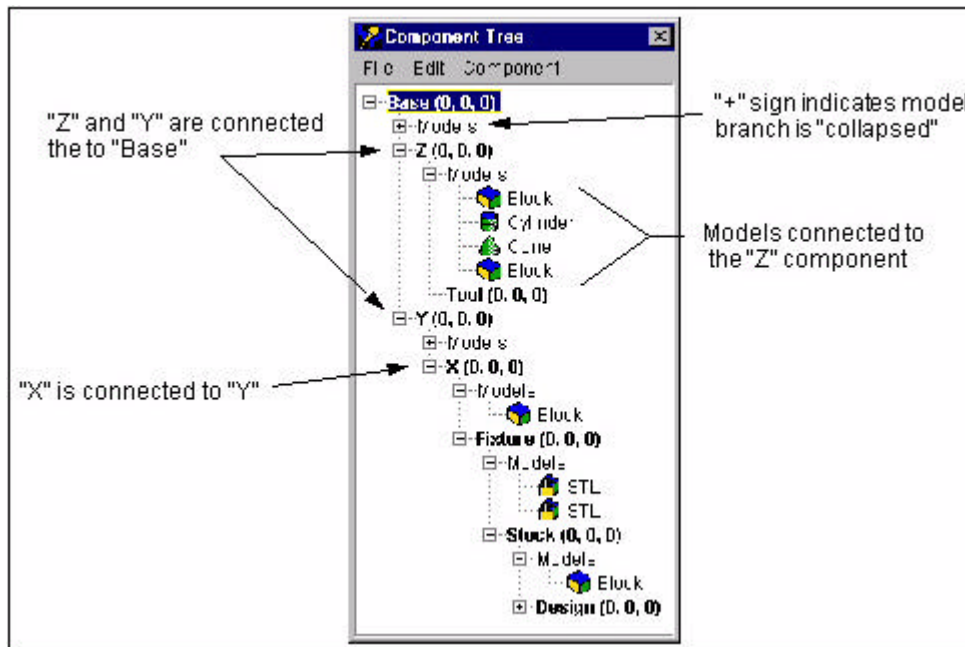
Components are connected to each other in a "tree" representative of how they are connected in the real world. The tree of components begins with a "Base", which is a non-moving component to which other components can be connected. Consider the following example: a stock workpiece is held for machining by a fixture. If the fixture is moved to a different location, the stock moves with it. To describe this in VERICUT using components, you would define a Fixture component, then add (append) a Stock component to the Fixture component.

The Component Tree window

The **Model menu > Component Tree** function opens the Component Tree window that shows the "tree" of components and models, and how they are connected. This window also provides a fast and easy graphical way of manipulating components and their models. Component names are listed in bold text with connect position values. Components and models added to a component are listed beneath the component name. Use the + / - symbols to expand or collapse branches of items in the tree.

Indentation indicates when items are connected to a parent component. If the parent component is moved or deleted, all items that are connected to the parent are also affected. In the sample window below if you deleted the "Fixture" component, the Fixture, Stock, and Design components AND their models would be deleted.

Sample Component Tree window:



Defining the tree of components

Each component has its own right-hand Cartesian coordinate system. To see axes representing a component's coordinate system (XcYcZc), select **View menu > Axes: Component**, open the Component Tree window, then select the component. Components are added, or "appended" to each other using the **Component menu Insert/Append** option lists to create the tree's branching structure.

Use features in the **Edit** menu to move, copy, or delete a selected component, as well as configure component attributes. These features are also available via right-clicking on a component name in the tree, and selecting from the displayed menu.

Once defined, you can move components using features on the **Model menu > Model Definition: Position** tab, or assemble with other components/models using **Assemble** tab features. Components can also be made invisible or visible again at any time during the simulation.

By default, components do not have size or shape. 3-D size and shape is provided via adding models to components, as described in the next section. While you are not obligated to add models to all components in a simulation, you will want to add them to support material removal, collision checking, and provide good visualization of the machining process.

Models

Add one or more models to a component to describe its 3-D size and shape. VERICUT provides interactive tools to define simple block, cylinder, or cone-shaped models. You can also import previously defined model files, such as: Stereolithography (STL), IGES, or VERICUT model files. See "File Types and Descriptions" in the *VERICUT Files* chapter for descriptions of these file types.

The Modeling window

The **Model menu > Model Definition** function opens the Modeling window that defines models or imports model files, and adds them to components. This multi-purpose window also orients components and models, as well as assigns component attributes. The window values are for the selected component or model in the Component Tree window. Changes made to values in the window affect the selected

Help on VERICUT

component (or model).

Defining models

Each model has its own right-hand Cartesian coordinate system. To see axes representing a model's coordinate system (XmYmZm), select **View menu > Axes: Model**, open the Component Tree window, then select the model. Use features on the **Model menu > Model Definition: Model** tab to define simple shapes or reference STL or VERICUT model files. Use **Import** tab features on this window to import an IGES file containing a solid or other 3-D entities. (2-D IGES files can not be used in VERICUT.)

Once defined, you can move models using features on the **Position** tab, or assemble with other components/models using **Assemble** tab features.

Tips:

Use the **Analysis menu > X-Caliper** function to measure and verify component/model location and orientation.

Features used to manipulate components are displayed in a shortcut menu when a component is selected and the right mouse button is clicked in the Component Tree window. Similarly, features used to manipulate models are displayed when a model is selected and the right mouse button is clicked.

Click repeatedly on a model in the graphics area to toggle selection between the model and its component. Refer to the Component Tree window to review what is selected.

Using the center mouse button of a three button mouse, click repeatedly on a model in the graphics area to toggle selection between different models along the line of sight from the pick point. Use this technique to select models hidden from view by an object in the foreground.

What do you want to do?

Component-related:

- Add, copy, or delete a component
- Specify component attributes
- Change component order in the tree
- Rename a component
- Change the component type
- Move a component or model

Model-related:

- Add, copy, or delete a model
- Import an IGES model file
- Move a component or model
- Set tolerance for modeling with revolved shapes
- Delete material
- Check stock model consistency

Working with components

Adding, copying, or deleting a component

A component defines the function of a solid object. Components are connected to each other in a "tree" representative of how they are connected in the real world. The Component Tree window shows the "tree" of components and how they are connected. This window also provides a fast and easy graphical way of manipulating components and their models. Use the + / - symbols to expand or collapse branches of items in the tree.

The procedures below describe a few of the many ways VERICUT allows you to add and manipulate components in the tree.

To add a component to the tree:

1. Click  on the Toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to add the new component to.

To add the new component before the selected component: click **Component menu > Insert > [component_type]**.

To add the new component after the selected component: click **Component menu > Append > [component_type]**.

VERICUT adds the new component to the tree.

3. Press **Reset**  after changing components in the component tree.

Use **Model menu > Model Definition** window features to add models or specify component attributes.

Shortcut: Right-click on the component in the tree, then choose the desired option from the menu that appears.

To copy and paste a component in the tree:

1. Click  on the toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to copy.
3. Click **Edit menu > Copy**. (**Note:** All connected components/models are also copied.)
4. Select the component to add the copied component to.
5. Click **Edit menu > Paste**.

VERICUT adds the copied components/models to the tree.

6. Press **Reset**  after changing components in the component tree.

Shortcut: Right-click on the component in the component tree, then choose the desired option from the menu that appears.

To delete a component from the tree:

1. Click  on the toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to delete. (**Note:** All connected components/models will also be deleted.)
3. Click **Edit menu > Delete**.
4. Click **Yes** to confirm deleting the component.
5. Press **Reset**  after changing components in the component tree.

Shortcut: Right-click on the component in the component tree, then choose the desired option from the menu that appears.

Tips:

Features used to manipulate components are displayed in a shortcut menu when a component is selected and the right mouse button is clicked in the Component Tree window. Similarly, features used to manipulate models are displayed when a model is selected and the right mouse button is clicked.

Click repeatedly on a model in the graphics area to toggle selection between the model and its component. Refer to the Component Tree window to review what is selected.

Using the center mouse button of a three button mouse, click repeatedly on a model in the graphics area to toggle selection between different models along the line of sight from the pick point. Use this technique to select models hidden from view by an object in the foreground.

Specifying component attributes

When you add a component VERICUT provides component attributes, such as: color, motion axis, rapid rate, etc. VERICUT handles components and their solid models differently depending on the component's type and attributes. Use the procedure below to specify component attributes as needed.

To specify component attributes:

1. Click  on the Toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to specify attributes for.
3. Click **Edit menu > Component Attributes**.
4. Configure the window as required. (See window features)
5. Press **OK** or **Apply**.
6. Press **Reset**  after changing components in the component tree.

Shortcut: Right-click on the component in the tree to specify attributes for, then choose the desired option from the menu that appears.

Changing component order in the tree

Components are connected to each other in a "tree" representative of how they are connected in the real world. The **Model menu > Component Tree** function opens the Component Tree window that shows the "tree" of components and models, and how they are connected. This window also provides a fast and easy graphical way of manipulating components and their models.

Use the procedure below to change the order of components in the tree. An example follows.

To change the order of components in the tree:

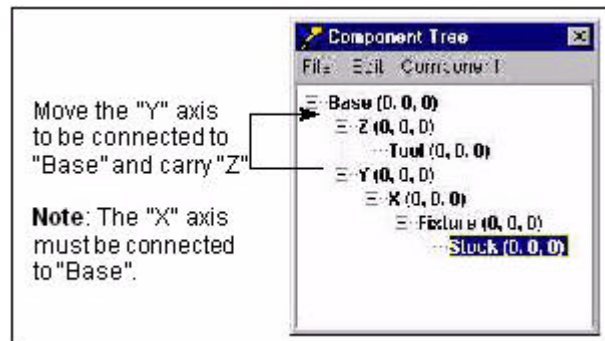
1. Click  on the Toolbar, or click **Model menu > Component Tree**.
 2. In the Component Tree window, select the component to move.
 3. Click **Component menu > Cut** (or **Copy**). (Note: All connected components/models are also cut.)
- VERICUT removes components/models cut from the tree.
4. Select to component to add the cut component to.
 5. Click **Component menu > Paste**.

VERICUT adds the components/models to the tree.

6. Press **Reset**  after changing components in the component tree.

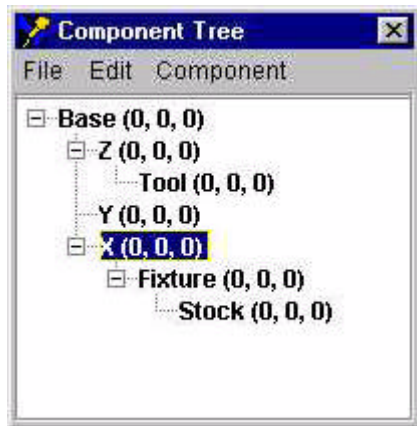
Shortcut: Right-click on the component in the tree to cut (or copy), then choose the desired option from the menu that appears.

Example- change the order of components in the machine below to move the "Y" component from the "stock-side" of the machine to be on the "tool-side" of the machine such that the Y-axis carries the Z-axis. This example makes use of "right-click" shortcut menus to accomplish this task quickly and easily.



To begin, cut "X" from "Y" and paste to "Base":

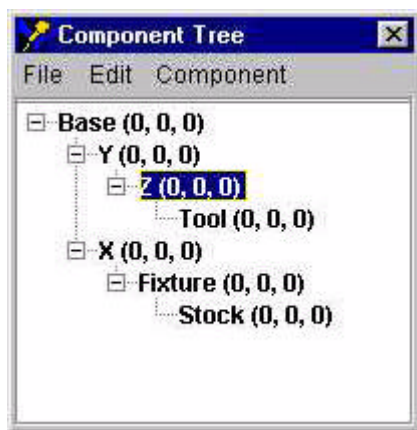
1. Right-click on the "X" component, select **Cut** from the shortcut menu.
2. Right-click on "Base", select **Paste** from the menu.



When pasted, the "X" branch of the tree is collapsed. Click the "+" sign to expand-repeat for the remaining components to see them all. Note that at this point "Z", "Y", and "X" are all connected to "Base".

Cut "Z" from "Base" and paste to "Y":

3. Right-click on "Z", select **Cut**.
4. Right-click on "Y", select **Paste**.



This is the desired configuration-you are done!

Of course, remember to press **Reset**  after changing components in the component tree.

Renaming a component

By default, component names are automatically supplied by VERICUT based on the component type. Use the procedure below to change the name of a component.

To rename a component:

1. Click  on the Toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to rename.
3. Click **Component menu > Rename**.
4. Type the new component name and press <Enter> (or <Return>).

Shortcut: Right-click on the component in the tree to rename, then choose the desired option from the menu that appears.

Changing the component type

When you add a component VERICUT automatically sets the component type. Use the procedure below to change a component's type.

To change a component's type:

1. Click  on the Toolbar, or click **Model menu > Component Tree**.
2. In the Component Tree window, select the component to change type.
3. Click **Edit menu > Component Type > [component_type]** where "component_type" represents the desired type.
4. Press **Reset**  after changing components in the component tree.

Moving a component or model

After adding a component or model, you can move the object using features on the **Position** or **Assemble** tabs in the **Model menu > Model Definition** window. Positioning supports moving the object via translation or rotation. Data is supplied via entering exact values, or graphical picking. Assembly supports moving the object by mating or aligning one to three model surfaces with surfaces on other models.

What do you want to do?

Translate a component or model via positioning

Rotate a component or model via positioning

Move a component or model via assembling

To translate a component or model via positioning:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the component or model in the Component Tree window to move. Click "+" to expand branches of the tree that you can not see.
2. Click  on the Toolbar, or click **Model menu > Model Definition**.
3. Ensure **Active Component** and **Selected** indicate the desired object to move.
4. Click the **Position** tab.
5. Click in the **From** field, then enter or pick a point on a model to represent the "move-from" point. (A crosshair and vector show you the pending pick-point location.)
6. Click in the **To** field, then enter or pick a point on a model to represent the "move-to" point. (Distance to translate is the incremental difference between the move-to and move-from points.)
7. Click **Move**.

To rotate a component or model via positioning:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the component or model in the Component Tree window to move. Click "+" to expand branches of the tree that you can not see.
2. Click  on the Toolbar, or click **Model menu > Model Definition**.
3. Ensure **Active Component** and **Selected** indicate the desired object to move.
4. Click the **Position** tab.
5. Click in the **Center of Rotation field**, then enter or pick a point on a model to rotate about.



To see the center of rotation, press . To remove the center of rotation symbol press this button again.

6. Enter the degrees of rotation in the **Increment** field.
7. Press a rotation button (**X+/X-**, **Y+/Y-**, **Z+/Z-**) to apply the incremental degrees of rotation.

To move a component or model via assembling:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the component or model in the Component Tree window to move. Click "+" to expand branches of the tree that you can not see.

2. Click  on the Toolbar, or click **Model menu > Model Definition**.
3. Ensure **Active Component** and **Selected** indicate the desired object to move.
4. Click the **Assemble** tab.
5. Click in the arrow by the first **Constraint Type**, then choose how to relate the objects being assembled:
 - **Mate**- normals of selected surfaces will oppose each other.
 - **Align**- normals of selected surfaces will point in the same direction.See "Model menu > Model Definition - Assemble tab" in the *Reference* chapter for examples of mated and aligned objects.
6. Click the surface on the object to move, then click the surface to which the object is to be mated (or aligned).

VERICUT shows the constraint is satisfied by highlighting it with a checkmark.

7. If desired, enter an **Offset** to apply to the constrained surfaces.

You can create up to three constraints using the above procedure.
8. Press **Apply** to move the object.

Working with models

Add, copy, or delete a model in a component

Features on the **Model menu > Model Definition: Model** tab are used to interactively define and add models to components. Models can be simple parametric shapes, or you can refer to external STL and VERICUT model files. The Component Tree window shows the "tree" of components and models, and how they are connected. This window also provides a fast and easy graphical way of manipulating components and their models. Use the + / - symbols to expand or collapse branches of items in the tree.

The procedures below describe a few of the many ways VERICUT allows you to add and manipulate models in components.

To add a model to a component:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the component to add the new model to.
2. Click  on the Toolbar, or click **Model menu > Model Definition**.
3. Click the **Model** tab.
4. Choose the desired model **Type** to add, then enter the supporting data to describe the model. (See window features)
5. Press **Add**.

VERICUT adds the model to the component.

Shortcut: Right-click on an existing model in the component tree, then choose the desired option from the menu that appears.

To copy and paste a model in a component:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the model to copy.
2. Click **Edit menu > Copy**.
3. Select to component to add the copied model to.
4. Click **Edit menu > Paste**.

Shortcut: Right-click on the model in the component tree, then choose the desired option from the menu that appears.

To delete a model from a component:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the model to delete.

Help on VERICUT

2. Click **Edit menu > Delete**.
3. Click **Yes** to confirm deleting the component.

Shortcut: Right-click on the model in the component tree, then choose the desired option from the menu that appears.

Tips:

Features used to manipulate components are displayed in a shortcut menu when a component is selected and the right mouse button is clicked in the Component Tree window. Similarly, features used to manipulate models are displayed when a model is selected and the right mouse button is clicked.

Click repeatedly on a model in the graphics area to toggle selection between the model and its component. Refer to the Component Tree window to review what is selected.

Using the center mouse button of a three button mouse, click repeatedly on a model in the graphics area to toggle selection between different models along the line of sight from the pick point. Use this technique to select models hidden from view by an object in the foreground.

Importing an IGES model file

Features on the **Model menu > Model Definition: Import** tab are used to import IGES model files for use as stock, fixture, design, or machine component models. An option is also available to create a solid model from an IGES surface. The surface can be used as is, or offset before converting it into a solid.

During import, VERICUT converts the IGES model data into a VERICUT Polygon file (.ply extension) and automatically refers the converted file.

To import an IGES model file:

1. Click  on the Toolbar, or click **Model menu > Component Tree**, then select the component to add the new model to.
2. Click  on the Toolbar, or click **Model menu > Model Definition**.
3. Click the **Import** tab.
4. Type or select the name of the **IGES File** containing the IGES data to convert.
5. Enter the **Tolerance** allowed from the IGES surface for surface approximation.
The steps 6-9 below apply only to converting an IGES surface into a solid:
6. Select **Offset Surface**.
7. Enter any **Offset Amount** to apply to the surface, for example to add stock for machining on the converted solid model.
8. Enter the **Base Location- Z** value of the plane to project the surface to.
9. To expand the surface edges to create a rectangular base, click **Extend to Rectangle**.
10. Press **OK** or **Apply**.

VERICUT converts the IGES model file to a VERICUT Polygon file and adds it to the active component.

Setting tolerance for modeling with revolved shapes

Use the procedure below for setting the tolerance for creating solid models from revolved shapes, such as cylinders, cones, and turning models created from IGES profiles.

To set the tolerance for creating solids from revolved shapes:

1. Click **File menu > Properties**.
2. Click the **Tolerance** tab.
3. Enter a **Cutting Tolerance** value appropriate for cutting your the workpiece. The value entered should also reflect adequate tolerance for analyzing the model after cutting.
4. Press **OK** or **Apply**.

Deleting material from the workpiece

VERICUT's **Model menu > Delete Material** function can delete or keep pieces of a model. This is particularly useful for deleting pieces of excess stock that have been cut free from the workpiece.

To delete (or keep) pieces of material:

1. Click  on the Toolbar, or click **Model menu > Delete Material**.
2. Select **Delete** or **Keep**, depending on the action desired.
3. Click on each piece of material to delete (or keep).

As each piece is selected, VERICUT erases the selected material from the view.

4. Press **OK** to approve deletion, or **Cancel** to restore the model pieces and start over.

Note: Once deleted, model pieces can not be restored.

Checking stock model consistency

Use the procedure below to check consistency of the stock models loaded for cutting.

To check the consistency of stock models:

1. Prior to cutting, click **File menu > Properties**.
2. Click the **General** tab.
3. Select **Stock Consistency Check**.
4. Press **OK** or **Apply**.

Tips:

This procedure is highly recommended for use on stock models creating from importing IGES data.

Integrating CAD/CAM data

Importing and exporting model and tool path data

VERICUT provides many methods of integrating data from CAD/CAM systems. Full CAD/CAM interfaces are offered that transfer all necessary data to VERICUT, as well as a suite of converters that transform geometry and tool path data from a variety of sources into data you can use in VERICUT or Stereolithography rapid-prototyping. Taking advantage of your CAD/CAM resources makes the simulation easier to setup, and more realistic.

The sections that follow describe ways you can import geometry and tool path data into VERICUT, or export the solid VERICUT machined workpiece. CAD/CAM data can be processed interactively, or via batch processing.

What do you want to do?

Conversion options:

- Convert IGES geometry data (See IGES data and version support info)

- Convert VDA geometry data (See VDA data and version support info)

- Repair STL or VERICUT model file normals

- Convert surface data to solid models

- Convert binary APT data

- Convert Mastercam NCI data

Import options:

- Import an IGES model file

Export options:

- Export models from VERICUT

Converting IGES data

Use the IGES Converter and the procedure below to interactively convert IGES data to VERICUT or STL model files. **Note:** IGES data can also be converted via batch processing.

To convert IGES data to VERICUT or STL model files:

1. Access the IGES Converter window: in VERICUT select **File menu > Convert > IGES**.

You can also run this converter outside of VERICUT via the "iges2v" command file located in the CGTech "commands" directory, under the directory representing your computer type.
2. Type or select the **Input File** containing the IGES data to convert.
3. Type or select the **Output File** to receive converted geometry.
4. Choose the **Output File Type** and **Output Format Type** in which to write the converted geometry.
5. If the **Output File Type= VERICUT**, choose the **Normal Direction** in which surface normal vectors point, as viewed in the CAD system. If unknown choose **INCONSISTENT**
6. Choose the desired **Surface Input Mode** (**ALL**—all surfaces, **BASE**—base surfaces only, **TRIMMED**—trimmed surfaces only)
7. Enter the **Tolerance** allowed from the IGES surface for surface approximation.
8. Click **Process**.

Error and informational messages from the process are sent to the IGES Converter window message area, and to a log file named "iges2v.log" created in the working directory.

Hints:

The IGES Converter does not convert 2-D geometry data, since this data is useless in VERICUT. Only 3-D surfaces and solids, and their supporting geometry is converted.

When expecting to use IGES data as solid models in VERICUT, ensure that surface normals point unidirectionally outwards in the CAD model prior to outputting the IGES data.

IGES data and version support

CGTech's IGES Converter converts the following data types conforming to the IGES 4.0 specification:

- 100: Circular arc
- 102: Composite curve
- 104: Conic arc (parabola, ellipse, hyperbola)
- 106: Copious data
- 108: Plane (bounded)
- 110: Line
- 112: Parametric spline curve
- 114: Parametric spline surface
- 116: Point
- 118: Ruled surface
- 120: Surface of revolution
- 122: Tabulated cylinder
- 124: Transformation Matrix
- 126: Rational B-spline curve
- 128: Rational B-spline surface
- 130: Offset curve
- 140: Offset surface
- 142: Curve on parametric surface
- 144: Trimmed parametric surface
- 150: Block
- 152: Right angular wedge
- 154: Right circular cylinder
- 156: Right circular cone frustum
- 158: Sphere
- 160: Torus
- 162: Solid of revolution
- 164: Solid of linear extrusion

Help on VERICUT

168: Ellipsoid

Note: The IGES Converter does not convert 2-D geometry data. Only 3-D surfaces, solids, and their supporting geometry is converted.

Converting VDA data

Use the VDA Converter and the procedure below to convert VDA data to VERICUT or STL model files, or Design Points files for use by the AUTO-DIFF function. **Note:** VDA data can also be converted via batch processing.

See also supported VDA data and version.

To convert VDA data to VERICUT or STL model files, or Design Points files:

1. Access the VDA Converter window: in VERICUT select **File menu > Convert > VDAFS**.

You can also run this converter outside of VERICUT via the "vda" command file located in the CGTech "commands" directory, under the directory representing your computer type.
2. Type or select the **Input File** containing the VDA data to convert.
3. Type or select the **Output File** to receive converted geometry.
4. Choose the **Output File Type** and **Output Format Type** in which to write the converted geometry.
5. If the **Output File Type= VERICUT**, choose the **Normal Direction** in which surface normal vectors point, as viewed in the CAD system. If unknown choose **INCONSISTENT**
6. Enter the **Tolerance** allowed from the VDA surface for surface approximation.
7. Click **Process**.

Error and informational messages from the process are sent to the VDA Converter window message area, and to a log file named "vda.log" created in the working directory.

Hints:

When expecting to use VDA data for solid models in VERICUT, ensure that surface normals point unidirectionally outwards in the CAD model prior to outputting the VDA data.

VDA data and version support

While CGTech's VDA Converter reads all VDA elements, the following data types conforming to the VDA 2.0 specification can be converted:

POINT

PSET (point set)

MDI (set of point/vectors)

SURF (base surface)

FACE (trimmed surface)

Repairing STL or VERICUT model file normals

Use the PolyFix Converter and the procedure below to repair surface normals of STL and VERICUT model files. **Note:** normals can also be repaired via batch processing.

Model files with inconsistent normals often appear OK in VERICUT until the solid model database is established, such as occurs when you start cutting or analyzing the model. Portions or the entire model may disappear, or verification results in the damaged areas may be questionable. PolyFix repairs the normals to ensure they point uniformly outward from the model surface.

To repair STL/VERICUT model file normals:

1. Access the PolyFix Converter window: in VERICUT select **File menu > Convert > PolyFix**.
You can also run this converter outside of VERICUT via the "polyfix" command file located in the CGTech "commands" directory, under the directory representing your computer type.
2. Type or select the **Input File** containing the normals to repair.
3. Type or select the **Output File** to receive converted geometry.
4. Choose the **Input File Type** appropriate for the model being repaired.
5. Choose the **Output File Type** and **Output Format Type** in which to write the repaired geometry.
6. Click **Process**.

Error and informational messages from the process are sent to the PolyFix window message area, and to a log file named "polyfix.log" created in the working directory.

Hints:

PolyFix-corrected model files can be used in VERICUT as solid stock, fixture, design, or NC machine shapes. It is a one time effort to repair an inconsistent model file.

After repair, use the "**Outward**" normals option when referring to the repaired model file in VERICUT.

Converting surface data to solid models

Use the Surface to Solid Converter and the procedure below to create solid models from open surface VERICUT or STL model files. This converter is particularly useful to create casting/forging stock models from design surface data. **Note:** surface data can also be converted to solid models via batch processing.

To convert surface data to a solid model file:

1. Access the Surface to Solid Converter window: in VERICUT select **File menu > Convert > Surface to Solid**.

You can also run this converter outside of VERICUT via the "sur2stk" command file located in the CGTech "commands" directory, under the directory representing your computer type.
2. Type or select the **Input File** containing the IGES data to convert.
3. Type or select the **Output File** to receive converted geometry.
4. Choose the **Input File Type** appropriate for the surface model being converted to a solid.
5. Choose the **Output File Type** and **Output Format Type** in which to write the solid model geometry.
6. Enter the **Tolerance** for offsetting and projecting the surface to create the solid.
7. Enter any **Offset** amount to apply to the surface, for example to add stock for machining on the converted solid model.
8. Enter the **Base Location- Z** value of the plane to project the surface to.
9. To expand the surface edges to create a rectangular base, click **Project Rectangle**.
10. To transform the solid model, enter 9 values separated by spaces in the **Transformation Matrix** field. Format: dx,dy,dz,Ivx,Jvx,Kvx,Ivy,Jvy,Kvy

where "dx,dy,dz" specify the distance to the local coordinate system origin, and "Ivx,Jvx,Kvx,Ivy,Jvy,Kvy" specify the orientation of the local coordinate system X and Y axis vectors with respect to the surface model coordinate system.

Tip: Matrix values correspond to MSYS record values that appear in a UG CLS file.

11. Click **Process**.

Error and informational messages from the process are sent to the Surface to Solid Converter window message area.

Converting binary APT data

Use the Binary CL Converter and the procedure below to convert binary APT CL data into ASCII low level APT tool path files. **Note:** binary APT data can also be converted via batch processing.

To convert binary CL data to ASCII APT tool path files:

1. Access the Binary CL Converter window: in VERICUT select **File menu > Convert > Binary CL**.

You can also run this converter outside of VERICUT via the "bincl2v" command file located in the CGTech "commands" directory, under the directory representing your computer type.

2. Type or select the **Input File** containing the binary CL data to convert.
3. Type or select the **Output File** to receive converted APT tool path records.
4. If required, type or select the **APT Table File** containing the desired blocking factor and binary data to APT word table.
5. Choose the **Input File Type** that best describes the source of the binary CL data.
6. If cutter height is not specified in the binary CL data, enter a **Minimum Cutter Height** value.
7. By default, circular motions will be output as CIRCLE records. To output GOTOs instead of CIRCLES, clear the **Output Circles** checkbox.
8. Click **Process**.

Error and informational messages from the process are sent to the Binary CL Converter window message area. The APT CL-record number (not ISN) is placed in columns 73 through 80 on each output record of the tool path. This number can be useful when using the **History** function to identify tool path records responsible for errors and other machine cuts. (Ref. **Analysis menu > X-Caliper**)

Tips:

If the output APT tool path file contains incorrect, little, or no APT record data, verify that the blocking factor is correct for the binary data. Typical blocking factors are: 512, 3228, 4628, 15476.

If the output APT tool path file has major or minor words other than expected, try editing the APT Table file to change the "Word" output for a specific APT class/subclass.

Converting Mastercam NCI data

The Mastercam NCI Converter and the procedure below to convert Mastercam NCI files to ASCII low level APT tool path files. **Note:** NCI data can also be converted via batch processing.

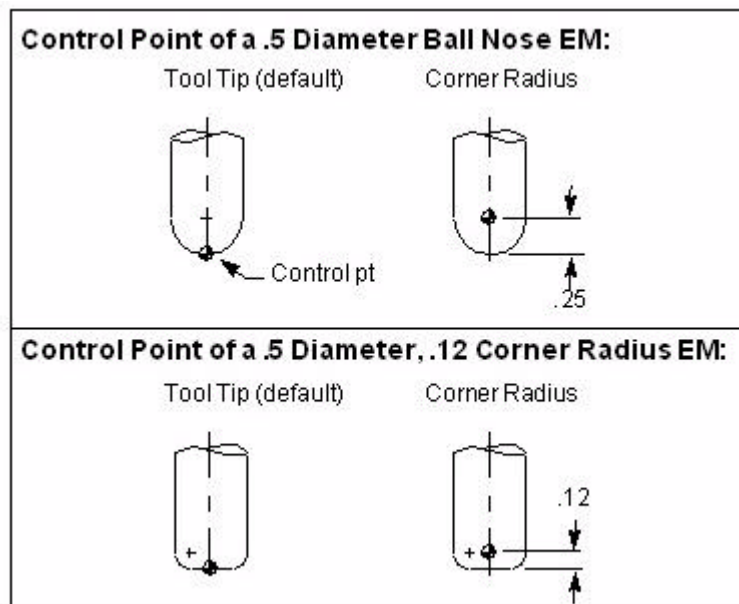
To convert NCI data to ASCII APT tool path files:

1. Access the Mastercam Converter window: in VERICUT select **File menu > Convert > Mastercam**.

You can also run this converter outside of VERICUT via the "mcam2v" command file located in the CGTech "commands" directory, under the directory representing your computer type.

2. Type or select the **Input File** containing the NCI data to convert.
3. Type or select the **Output File** to receive converted APT tool path records.
4. Enter the desired **Cutter Height** to apply to all cutting tools.
5. Enter the chordal **Tolerance** for approximating circular motions with chordal GOTO points.
6. Enter the **Tool Change Retract Height**-Z value of the plane where the cutter is moved to for tool change (with respect to the tool path coordinate system).
7. Choose the **Control Point** option which best describes the driven point of the cutter. See the examples that follow.

Examples of setting the **Control Point**:



8. Click **Process**.

Error and informational messages from the process are sent to the Mastercam Converter window message area.

Exporting models from VERICUT

Using the Model Export option, VERICUT supports exporting the cut VERICUT model in a variety of file types. Use the procedures below to export the model, including all machine cuts, to an STL, VERICUT Polygon, or IGES file. When done, a message informs you how many IGES entities, or STL/VERICUT Polygon triangles were created.

Unless otherwise specified, the procedures below assume the VERICUT model was cut in "**Standard**" cut mode. (Ref. **File menu > Properties, General** tab.)

Export a cut model to an IGES file

Export a cut model to an STL or VERICUT Polygon model file

Export a FastMill cut model

To export a cut model to an IGES file:

1. After cutting, click **File menu > Export Model**.
2. Type or select the name of the **File** to receive exported geometry.
3. Choose **File Type=IGES**.
4. Choose the **Model Coordinates** (coordinate system) and **Units** in which to export the model data.
5. Choose the **Process** option corresponding to the desired method of processing VERICUT model geometry into IGES data. **Machined Features**—recommended for most parts with distinct features, such as: pockets, ribs/walls, etc. **Grid**—recommended for mold and die type parts where machined features are primarily blended or sculptured surfaces, etc.
6. If processing via "**Grid**", enter the number of grid squares to spread over the model surface. More squares = more accuracy and larger output file size (60 max).
7. Set **Connect angle** to specify the angle between cut model features to group into an IGES surface.
8. By default, **NURBS Only** is selected active and outputs cut model data in NURBS data entities (recommended for IGES data destined for Pro/Engineer). Clear this checkbox to output IGES data in any entity format.
9. If desired, select **Minimum Edge** and specify the minimum length (in units) of a boundary or "trimming" curve segment. Clear this checkbox to output IGES trimming curves as formed. (Can cause VERICUT to output IGES data too small for your CAD system to process.)
10. Click **Output**.

Back to top

To export a cut model to an STL or VERICUT Polygon model file:

1. After cutting, click **File menu > Export Model**.
2. Type or select the name of the **File** to receive exported geometry.

Help on VERICUT

3. Choose **File Type**=STL or **VERICUT** and **File Format** in which to write exported geometry.
4. Choose the **Model Coordinates** (coordinate system) and **Units** in which to export the model data.
5. Click **Output**.

Back to top

To export a FastMill cut model:

1. Ensure the FastMill cut model is displayed with the desired **Display Resolution**, as this affects output model accuracy. (Ref. **File menu** > **Properties, General** tab.)
2. Click **File menu** > **Export Model**.
3. Type or select the name of the **File** to receive exported geometry.
4. Choose the **File Type** and **File Format** in which to write exported geometry.
5. Choose the **Model Coordinates** (coordinate system) and **Units** in which to export the model data.
6. Set **Output Resolution** and **Output Precision** as required to provide adequate quality and file size of the exported model.
7. Click **Output**.

Back to top

Configuring for tool path simulation

Specifying the tool path file(s) to simulate

Use the procedures below to configure VERICUT to simulate one or more tool path files. Different procedures are provided for specifying a single tool path file, or multiple tool paths:

To specify a single tool path file to simulate

To specify multiple tool path files to simulate

Tip: After specifying tool paths to simulate, be sure to set a **Cutting Tolerance** that is appropriate for simulation speed and analyzing the model after cutting. (**File menu** > **Properties: Tolerance** tab)

To specify a single tool path file to simulate:

1. Click  on the Toolbar, or click **Setup menu** > **Toolpath**.
2. Choose the **Toolpath Type** which best describes your tool path file. **Note:** G-Code Data tool path types may additional configuration-see "Simulating G-code tool paths" in this chapter.
3. Select the current tool path file name to replace.
4. Type the new tool path name, or click **Replace** and select the tool path file to simulate.
5. Use the "Current" coordinate system, or left-click on "Current" and in the **Coord. System** list, then choose the coordinate system that orients your tool path for proper machining. If the coordinate system is not defined, access **Setup menu** > **Coord. System** to define it, then return to this window to associate it with the tool path.

While you are in this window, you may also want to configure how VERICUT will receive cutting tool descriptions-ref. "Supply cutting tool descriptions" in this chapter.

6. Click **OK** or **Apply**.
7. If there isn't anything in the tool path file identifying the unit coordinate system AND tool path units do not match the session units, then set default tool path units as follows:
 - Click **Setup menu** > **APT Settings**.
 - Click the **Motion** tab, set **Default Toolpath Units** as needed, then click **OK**.

To specify multiple tool path files to simulate:

This procedure is similar to that for specifying a single tool path above (please review).

1. Click  on the Toolbar, or click **Setup menu** > **Toolpath**.
2. Choose the **Toolpath Type** which best describes your tool path file.
3. If the list contains tool path files that are not wanted, click **Clear** to clear all tool path file names

from the list. (**Note:** Do not clear tool path files that have already been simulated—they will be needed if you reset and recut the model.)

4. Click **Add**.
5. In the window that opens, select tool path files to simulate. To select multiple tool path files in this window, try these techniques:
 - *Select multiple files in sequence*—Click first file name in the sequence, then press and hold the <Shift> key while clicking the last file name.
 - *Select additional files*—Press and hold the <Control> key while selecting each additional file. Selecting twice unselects the file.
6. When the desired tool paths are selected, click **OK**. (Or repeat these steps until all desired tool paths are added.
7. Tool paths are processed by VERICUT in listed order. If necessary, reorder the items in the list via left-dragging the button at the start of the entry in the list.
8. Use the "Current" coordinate system, or left-click on "Current" and in the **Coord. System** list, then choose the coordinate system that orients each tool path for proper machining. If the coordinate system is not defined, access **Setup menu > Coord. System** to define it, then return to this window to associate it with the tool path.
9. Click **OK** or **Apply**.

Tip: Be sure to set a **Cutting Tolerance** that is appropriate for simulation speed and analyzing the model after cutting. (**File menu > Properties: Tolerance** tab)

Supplying VERICUT with descriptions of cutting tools

VERICUT offers a variety of ways to supply descriptions of cutting tools. Options are available to use cutter descriptions present in the tool path file, or retrieve tools defined in a VERICUT Tool Library file.

What do you want to do?

- Change tools with tool path cutter descriptions
- Change tools with tool or pocket numbers (requires a Tool Library)
- Change tools with tool path file names (requires a Tool Library)
- Change tools with program stop (requires a Tool Library)
- Set an initial tool (requires a Tool Library)
- Override the current tool (requires a Tool Library)
- Use a tool list (requires a Tool Library)
- Use tool holders (typically retrieved from a Tool Library)

Note: For options that use a Tool Library, see "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

Changing tools with tool path cutter descriptions

Use the procedure below to have VERICUT change tools based on processing parametric cutter descriptions found in the tool path file. Examples of such records include: CUTTER, TLDATA, or PPRINT/VERICUT-TC.

To change tools with cutter description records:

1. Click  on the Toolbar, or click **Setup menu > Toolpath**.

2. Choose **Tool Change By=Cutter Desc**.

Note: If cutter descriptions in the tool path file do not reflect true cutter shape, then choose **Tool Change By=List**. Refer to "Use tool lists" in this chapter for instructions on cross-referencing tool path cutter descriptions to tools having the correct shape in a Tool Library.

While you are in this window, you may also want to specify the tool paths for VERICUT to simulate-ref. "Specify the tool path file(s) to simulate" in this chapter.

3. Click **OK** or **Apply**.

Changing tools with tool or pocket numbers

Use the procedure below to have VERICUT change tools based on processing tool path records that reference tool or pocket numbers and retrieve associated tools from a Tool Library file. Examples of such records include: TnM6 (G-code tool paths), LOADTL or LOAD TOOL, or TURRET.

Note: This option requires using a Tool Library. See "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

To change tools with tool/pocket number records:

1. Click  on the toolbar, or click **Setup menu > Toolpath**.
2. Choose **Tool Change By=Tool Number**.

Note: If tool/pocket numbers do not match the desired IDs in the Tool Library, then choose **Tool Change By=List**. Refer to "Use tool lists" in this chapter for instructions on cross-referencing tool/pocket numbers to IDs of tools stored in a Tool Library.

While you are in this window, you may also want to specify the tool paths for VERICUT to simulate-ref. "Specify the tool path file(s) to simulate" in this chapter.

3. Click **OK** or **Apply**.

Changing tools with tool path file names

Use the procedure below to have VERICUT change tools based on tool path file names. With this option, you assign a tool from a Tool Library file to each tool path file. The tool is changed at the start of processing the tool path file.

Note: This option requires using a Tool Library. See "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

To change tools with tool path file names:

1. Click  on the toolbar, or click **Setup menu > Toolpath**.
2. Choose **Tool Change By=File Name**.

While you are in this window, you may also want to specify the tool paths for VERICUT to simulate-ref. "Specify the tool path file(s) to simulate" in this chapter.

3. For each tool path file in the list, type or left-click the **Tool ID** option list to assign the tool to use.
4. Click **OK** or **Apply**.

Changing tools with program stop (manual tool change)

Use the procedure below to have VERICUT change tools based on processing "program stop" records in the tool path. Examples of such records include: M0 (G-code tool paths) or STOP. With this option, you can simulate manual tool changes by creating a sequential list of stop records found in the tool path(s) to simulate, then assigning a tool from a Tool Library file to each "stop" tool change event.

Note: This option requires using a Tool Library. See "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

To change tools with program stop records:

1. Click  on the toolbar, or click **Setup menu > Toolpath**.
2. Choose **Tool Change By=List**.
3. Click **Use Tool List** to open the Tool Change List window.
4. Choose **List Tool Change By=Prog. Stop**.
5. Press **Build Tool List** to have VERICUT search the tool path(s) listed for simulation and build a sequential list of program stop records.
6. For each tool change event in the list, type or left-click the **Cutter ID** option list to assign the tool to use. Repeat for **Holder ID** if also using holders.
7. Click **OK** to close the Tool Change List window.

With the Toolpath window is still open, you may also want to specify the tool paths for VERICUT to simulate-ref. "Specify the tool path file(s) to simulate" in this chapter.

8. Click **OK** or **Apply**.

Tip:

Left-click in the Tool Change List window to quickly add and delete tool change events in the list.

Setting an initial tool

Use the procedure below to have VERICUT load an initial tool from a Tool Library file. Setting an initial tool is required when simulating tool paths for machines that have a tool loaded at the time tool processing is started, therefore do not have information about the first tool.

Note: This option requires using a Tool Library. See "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

To set an initial tool:



1. Click  on the toolbar, or click **Setup menu > Toolpath**.
2. Select **Initial Tool**.
3. Choose the ID of desired tool using the option list next to this feature.
4. Click **OK** or **Apply**.

Overriding the current tool

Use the "**Tool Override**" feature and the procedure below to override the current tool in use by VERICUT with a tool from a Tool Library file.

Note: This option requires using a Tool Library. See "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

To override the current tool:

1. Click  on the toolbar, or click **Setup menu > Toolpath**.
2. Select **Tool Override**.
3. Choose the ID of desired tool using the option list next to this feature.
4. Click **OK** or **Apply**.

The next motion is simulated using the override tool. VERICUT uses the override tool until you select a different tool ID or clear the **Tool Override** checkbox. When cleared, the tool that would have been in use becomes the active tool.

Using tool holders

The "**Use Holders**" feature controls when VERICUT displays and detects potential collisions with tool holders. The most common method of supplying VERICUT with tool holder descriptions is via a Tool Library file, as demonstrated by this procedure. However, holders can also be described using special tool path record formats. See "Defining profile tool shapes" in this chapter for more information.

You can turn holder display on or off at any time during the simulation using the procedure below.

To turn tool holder display on (or off):

1. Access **Setup menu > Tool Manager** and define the tool holders to use-ref. "Create a new Tool Library" in this chapter for information about defining a library of tools/holders.
2. Click  on the toolbar, or click **Setup menu > Toolpath**.
3. Select **Use Holders** to display tool holders and detect potential collisions, or clear this checkbox to turn holder display off.
4. Click **OK** or **Apply**.

Using tool lists

VERICUT can change tools based on a tool change event list, or "tool list" for short. You can use a tool list to specify the tools to use with sequential tool changes, or to replace tools specified for use in the tool path file. All tool lists contain references that link tool change events to retrieve tools stored in a Tool Library file-ref. "Create a new Tool Library" in this chapter for information about defining a library of tools referenced by tool changes.

Conditions that require using a tool list:

Tools manually changed when "program stop" codes are processed-these tool paths typically do not have tool descriptions or references to IDs or pocket numbers, except possibly in comments.

Tools changed via processing parametric cutter description records in the tool path file, however, descriptions do not reflect true cutter shape.

Tools changed via processing tool path records that reference tool or pocket numbers, however, pocket numbers do not match the desired IDs of tools in the Tool Library.

Using OptiPath to optimize cutting sequences via specifying optimization records for use at tool change events.

What do you want to do?

Build a new tool list

Add, modify, or delete tool list events

Override a tool in use from a tool list

Building a new tool list

Tool lists can be manually built via adding references to an empty list. However, the **Build Tool List** feature provides a fast and easy way to automatically build a tool list based on the active **List Tool Change By** option, and current list of tool path files to simulate. The procedure below uses this method.

To build a tool list:

1. Click  on the Toolbar, or click **Setup menu > Toolpath**.
2. Specify the tool path file(s) to simulate- ref. "Specify the tool path file(s) to simulate" in this chapter.
3. Choose **Tool Change By=List**.
4. Click **Use Tool List** to open the Tool Change List window.
5. Choose the **List Tool Change By** option to generate the tool list from. (See option descriptions)
6. To have VERICUT omit subsequent cutter descriptions with the same shape, select **Skip Duplicate Cutter Descriptions**. (Recommended for most APT-CLS tool path files.)
7. Press **Build Tool List** to have VERICUT search the tool path(s) listed for simulation and build a list of tool change events.
8. For each tool change event in the list, type or left-click the **Cutter ID** option list to assign the tool to use. Repeat for **Holder ID** if also using holders.
9. Click **OK** to close the Tool Change List window.
10. In the Toolpath window, click **OK** or **Apply**.

Tips:

Left-click in the Tool Change List window to quickly add and delete tool change events in the list.

See "Change tools with program stop" in this chapter for another example of using a tool list.

Adding, modifying, or deleting events in a tool list

Once a tool list is built, it handles all tool change activity during the simulation. If edits are made to tool path file(s), it may be necessary to edit the tool list. Use the procedures below to add, modify, or delete tool change events in the tool list for proper tool change activity.

To add a tool change event to the tool list:

1. Click  on the Toolbar, or click **Setup menu > Toolpath**.
2. Click **Use Tool List** to open the Tool Change List window.
3. Click **Add**.

A new "blank" tool change event is added.
4. In the **Event** field, type or left-click the EXACT text corresponding to the new tool change event as it appears in the tool path file. (Cut and paste is recommended to avoid typing errors.)
5. Type or left-click the **Cutter ID/Holder ID** to assign to the new event.
6. If necessary, rearrange tool change events via left-clicking and dragging events to desired positions in the list.
7. Click **OK** to close the Tool Change List window.
8. In the Toolpath window, click **OK** or **Apply**.

To modify an existing tool change event in the tool list:

1. Click  on the Toolbar, or click **Setup menu > Toolpath**.
2. Click **Use Tool List** to open the Tool Change List window.
3. Select the tool change event to modify.
4. Edit the tool change event as required.
5. Click **OK** to close the Tool Change List window.
6. In the Toolpath window, click **OK** or **Apply**.

To delete a tool change event from the tool list:

1. Click **Setup menu > Toolpath**.
2. Click **Use Tool List** to open the Tool Change List window.
3. Select the tool change event to delete.
4. Click **Delete**. (Use **Clear** to clear all tool change events.)

The tool change event is deleted from the list.

Help on VERICUT

5. Click **OK** to close the Tool Change List window.
6. In the Toolpath window, click **OK** or **Apply**.

Defining a coordinate system (orient tool path)

You control the coordinate system(s) in which VERICUT simulates tool path motions, measures, and sections models. A user-defined coordinate system is typically used to locate an APT or CLS tool path for proper relationship to the workpiece. However, you can also instate different coordinate systems for defining section planes or gathering measurement data.

Note: Coordinate systems are not recommend to locate G-code tool paths on a simulated 3-D machine, as this can cause the workpiece to move in the machine view. Instead, use an Input Program Zero table, or work offsets if present in the tool path file.

To define a new coordinate system:

1. Click  on the Toolbar, or click **Setup menu > Coord. System**.
2. Click on the **Coordinate System Name** option list, then enter a new coordinate system name.
3. Use one of the methods below to move the coordinate system from its default location:
 - *Graphical orientation*—Choose one of the **Orient** methods, then click the arrow and follow the prompts in the message area to define a coordinate system via picking three points, X and Y axes, or three planes.
 - *Translate*—Click in the **From** field, then enter or pick a point on a model to represent the "move-from" point. (A crosshair and vector show you the pending pick-point location.) Click in the **To** field, then enter or pick a point on a model to represent the "move-to" point. (Distance to translate is the incremental difference between the move-to and move-from points.) Click **Move**.
 - *Rotate*—Click in the **Center of Rotation field**, then enter or pick a point on a model to rotate about. To see the center of rotation, press . (To remove the center of rotation symbol press this button again.) Enter the degrees of rotation in the **Increment** field. Press a rotation button (**X+/X-**, **Y+/Y-**, **Z+/Z-**) to apply the incremental degrees of rotation.
4. Click **OK** or **Apply**.

VERICUT adds the coordinate system to the list. The coordinate system active when the window is closed is immediately instated for reference by tool path motions, X-Caliper measurements, and sectioning.

Note: Coordinate systems specified in the **Setup menu > Toolpath: Coord. System** list override the active user-defined coordinate system.

Choosing the machining type

The active machining type determines how VERICUT simulates machining the workpiece, and what types of cutting tools will be required.

To choose the machining type to simulate:

1. Click  on the toolbar, or click **Setup menu > Motion**.
2. Choose the desired **Machining Type**.
3. Press **OK** or **Apply**.

Tips:

This value can be set automatically via tool path comment records. APT-CLS example:
"PPRINT/VERICUT-MODAL MACHTYPE=milling". See "PPRINT" in Supported APT-CLS records section of the *VERICUT Files* chapter for more information.

Choosing the cut mode

The active cut mode determines if animated cutting motion is seen, and dramatically affects the time required to cut the workpiece. Availability and accuracy of some VERICUT functions are also affected. See **File menu > Properties** in the *Reference* chapter for details.

To choose the cut mode:

1. Prior to cutting, click **File menu > Properties**.
2. Click the **General** tab.
3. Select the desired **Cut Mode**:

Standard—Recommended for animated cutting simulation and highest degree of accuracy.

FastMill—Recommended to process large tool paths very quickly and without animation-good for typical automotive or mold and die-type parts.

4. Enter a **Model Resolution** value appropriate for your cutting application.
5. Press **OK** or **Apply**.

Tips:

Toolbar short cuts for cut modes and display resolution: 

Animation speed in standard cut mode is controlled via the **Setup menu > Motion: Animation Speed** feature.

Setting the tolerance for cuts in the workpiece

Use the procedure below for setting the tolerance for cuts applied to the workpiece. Increasing this number increases the speed of the simulation, and requires less system resources and space to store the model database. However, the accuracy of cuts in the model is also reduced. The value entered should reflect the accuracy needed for analyzing the model after cutting.

Tip: Best results are obtained when VERICUT is fully configured to simulate cutting the entire part before setting the **Cutting Tolerance**, e.g. all tool paths, all cutting tools that will be used, etc.

The **Cutting Tolerance** can only be set prior to cutting. If the model has been cut, reset the model to access this feature. (Press  on the VERICUT main window.)

To set the tolerance for cuts machined in the workpiece:

1. Click **File menu > Properties**.
2. Click the **Tolerance** tab.
3. Enter a **Cutting Tolerance** value appropriate for cutting your the workpiece. The value entered should also reflect adequate tolerance for analyzing the model after cutting.
4. Press **OK** or **Apply**.

Setting the tolerance for interpolated motions

Use the procedure below for setting the tolerance used to interpolate NURBS motions, machine positions during circle/arc and helical motions, and 4-axis positioning moves simulated via the **Detect 4-Axis Rotary Motion** feature.

To set the tolerance for interpolated motions:

1. Click **File menu > Properties**.
2. Click the **Tolerance** tab.
3. Enter a **Interpolation Tolerance** value appropriate for simulating the motions described above.
4. Press **OK** or **Apply**.

Setting a maximum safe cutting feed rate

The Fast Feed value in VERICUT establishes a feed rate threshold at which material removal is unsafe-likely to cause tool breakage, machine malfunction, or part damage. Material removed under these conditions is shaded in the **Error** color (typically red) and causes a “Fast feed rate” error.

To specify the feed rate threshold at which material removal is unsafe:

1. Click  on the toolbar, or click **Setup menu > Motion**.
2. Enter the **Fast Feed** value that is considered unsafe for cutting.
3. Press **OK** or **Apply**.

Tips:

You may want to adjust this value per tool. This can be done automatically via tool path comment records. APT-CLS example: "PPRINT/VERICUT-MODAL FASTFEED=100". See "PPRINT" in Supported APT-CLS records section of the *VERICUT Files* chapter for more information.

Controlling canned cycle simulation

VERICUT provides an easy way to control simulation of canned cycles, e.g. G8n, “CYCLE”, etc. Options enable you to see the full pecking cycle, plunge once to full depth, or ignore cycle motions completely.

To simulate or ignore canned cycle motions:

1. Click  on the Toolbar, or click **Setup menu > Motion**.
2. Choose **Drill Cycle=Full Motion** to see pecking, **Bottom Only** to simulate one full depth peck, or **Ignore** to ignore cycle motions.
3. Press **OK** or **Apply**.

Tips:

While VERICUT simulates most popular cycle motions, there are also capabilities to add more and change how existing cycles are interpreted- see "About configuring for cycle simulation" in the *Using VERICUT* chapter for more information.

Calculating minimum cutter extension

When simulating in FastMill cut mode with tool holders displayed, you can have VERICUT calculate the minimum cutter extension required to avoid shank/holder collisions. See the tip below for calculating minimum cut height required for a given cut sequence.

To calculate minimum cutter extension:

1. Assuming VERICUT is set to simulate in FastMill cut mode and tool holders are defined, ensure **Use Holders** is active to display them in the simulation. (Ref. **Setup menu > Toolpath**)
2. Click  on the toolbar, or click **Setup menu > Motion**.
3. Select **Calculate Min. Cutter Extension**.
4. To specify additional holder clearance, enter this value in the **Holder Clearance** field.
5. Press **OK** or **Apply**.
6. Cut the workpiece.

Minimum cutter extension calculations begin with the defined cutter height. If potential holder collisions are detected during the simulation, VERICUT extends the cutter to avoid the collision and adds this amount to the calculated height.

Note that when these calculations are being performed, tool holder collisions are not reported since VERICUT extends the cutter to avoid the collision.

To view calculated minimum cutter extension values:

1. Click **Info menu > VERICUT Log**.
A VERICUT Log window opens containing the Log file.
2. Review minimum cutter extension values listed under the column titled "Min" in the Tool Summary section.

Tips:

To have VERICUT calculate the minimum cut height required for a tool in a given cut sequence, try defining the cutter for simulation with a very small height value.

Simulating APT-CLS tool paths

About configuring for cycle simulation

By default, VERICUT simulates the motions of popular canned cycles, also known as "fixed tool axis cycles" found in APT or CLS tool paths. If needed, VERICUT also enables you to add more and change how existing cycles are interpreted. To do so, it must have access to cycle definitions, which are retained in the User file. The distributed default User files (vericut.usr, vericutm.usr, init.usr and initm.usr) contain cycle definitions that pertain to each of the supported flavors of APT canned cycles. The cycle definitions for one flavor of APT are referred to as a cycle definition set, or more simply, a cycle set.

You can adjust the definitions of the distributed cycles, add definitions to the sets, or provide different sets of your own using features located on the **Setup menu > APT Settings** window "**Cycles**" tab.

The three parts of a cycle definition

So what does a cycle definition look like? Let's examine an example from the default User files:

```
CYCLE/BORE,#fedto,#fedupm,MMPM,#rapto
```

```
MODALS:retract_mode=fedrat
```

```
USEWITH:VERICUT,CADRA,CV
```

A cycle definition can have three parts, the CYCLE statement, optional MODALS and USEWITH lines. This example is a member of three sets, for VERICUT, CADRA and CV flavors of APT, but more typically there would be just one flavor in the USEWITH line.

This document will explore each of the three parts of a cycle definition in detail. Let's refer to them as:

Cycle template

Modal specification

Applicability"

Cycle template

A cycle template appears similar to the CYCLE record to simulate. Numeric values in the statement that are relevant to the anticipated motion are replaced by mnemonics in the template. The major and minor words can simply be reproduced. For example, the following CYCLE record:

```
CYCLE/DRILL,40,200,MMPM,5
```

can be represented by the following cycle template:

```
CYCLE/DRILL,#fedto,#fedupm,MMPM,#rapto
```

Note that mnemonics all start with a "#" character to denote a numeric value. There are a limited number of valid mnemonics, with predefined and rigid interpretations. A description of each mnemonic is provided

Help on VERICUT

under the heading "Mnemonics for cycle templates" later in this section.

A cycle statement may contain numeric values which are not covered by any of the mnemonics, and which do not affect simulated cycle motion, for example- dwell time. Such values can be represented in the template by a "#" character without a mnemonic. Think of them as placeholders for the values. For example:

```
CYCLE/REAM,60,5,1,1,5,200,10,ON
```

can be represented by:

```
CYCLE/REAM,#fedto,#rapto,#,#,#rtrcto,#fedupm,#,A
```

The placeholders enable VERICUT to extract numeric values that affect the motion, in this case represented by #fedto, #rapto, #rtrcto and #fedupm, and ignore values that are not significant for simulating cycle motions.

In the above example, note that the minor word "ON" is not replicated in the template of this example. A similar ream CYCLE record could have the minor word "OFF". Rather than having two templates that only differ by the last minor word, we can use a placeholder for a word and cover the variations with a single template. "#" is a placeholder for a numeric value. You can use "A" or any word as a placeholder for a minor word that is not significant for simulating cycle motions.

But, there are two situations in which minor words **must** be replicated in the template.

1. If a minor word precedes the first numeric value, it must appear in the template explicitly. So in the prior example, "REAM" is replicated.
2. If a minor word is paired with a mnemonic, it must appear in the template.

For example:

```
CYCLE/BORE,FEDTO,60,MMPM,200,RAPTO,5
```

should be represented by:

```
CYCLE/BORE,FEDTO,#fedto,MMPM,#fedupm,RAPTO,#rapto
```

because each minor word after "BORE" is paired with a corresponding mnemonic. This enables VERICUT to process cycle statements with the same minor words in a different order, without requiring extra templates. Thus:

```
CYCLE/BORE,FEDTO,60,RAPTO,5,MMPM,200
```

would be covered by the same template as the prior example. Usually word/value pairs are in that order, but a feed rate and its units can be reversed. To phrase the rules for minor words another way, it is safest to replicate all minor words in the cycle template unless there are alternatives to a word. If there are alternatives and they precede all values, or are paired with a mnemonic, then it is necessary to enumerate the alternatives in several templates.

Minor word synonyms

Minor words for the units of feed rate are interchangeable, or "synonyms". Thus if you process APT files with imperial units and other APT files with metric units, you do not need multiple templates such as:

```
CYCLE/BORE,FEDTO,#fedto,IPM,#fedupm,RAPTO,#rapto
```

```
CYCLE/BORE,FEDTO,#fedto,MMPM,#fedupm,RAPTO,#rapto
```

Help on VERICUT

Either of these templates will suffice for either unit.

Within cycle templates the following minor words are synonyms:

IPM—inches per minute

MMPM—millimeters per minute

FPM—feet per minute

IPR—inches per revolution

MMPR—millimeters per revolution

FPR—feet per revolution

TPI—threads per inch

TPMM—threads per millimeter

Adding a minus sign to a mnemonic

Normally mnemonics consist of the character "#" and a word. But there can be a "-" sign between the "#" and the word if the corresponding value in a cycle statement is to be negated. This will handle the case that some post-processors expect a positive **fedto** ("feed to") value, while others expect a negative value for this action. By default, VERICUT assumes a positive **fedto** value will move toward or into the workpiece. While the "-" sign can be applied to any mnemonic, it is only likely to be used with **fedto** or **fedlist**.

Mnemonics for cycle templates

Following is an explanation of each mnemonic, in alphabetic order:

cskdia—Countersink diameter. Typically this mnemonic appears with another, **tlangl** and perhaps a third, **holdia**. If no tool angle is provided, a value of 118 degrees is used. If a hole diameter is provided then an initial rapid motion can be adjusted to place the tool partially within the hole prior to cutting. It doesn't make much sense to have a **fedto** distance with a countersink diameter. If one is provided, it will be ignored.

fedto—Feed depth from the cycle's datum (typically a GOTO point). Use "**#fedto**" if a positive value implies motion into the material. Use "**#-fedto**" if a negative value implies motion into the material.

fedlist—A list of increasing feed depths. A single mnemonic of this type represents any number of values in the cycle statement. For example if you have a statement like:

CYCLE/DEEP,DEPTH,10,20,25,MMPM,100

the template could be:

CYCLE/DEEP,DEPTH,#fedlist,MMPM,#fedupm

and it would handle statements with any number of depths between the minor words "DEPTH" and "MMPM" (or one of its synonyms). Use "**#fedlist**" if a positive value implies motion into the material. Use "**#-fedlist**" if a negative value implies motion into the material.

fedupm—Feed rate in units determined by the associated minor word in the cycle statement (if any).

holdia—Hole diameter. Only meaningful if a **cskdia** is also present.

rapto—Rapid clearance distance above the cycle's datum. The tool rapids to this clearance location prior to executing the cycle motion. If a retract distance (**rtreto**) is not provided, this value will also serve as the retract clearance distance.

rtreto—Retract distance above the cycle's datum. If the cycle involves multiple retract motions, such as a pecking operation, all the retracts may or may not be to this height, but the last one always will be.

step—An incremental depth by which each successive feed in a pecking operation will increase. It is possible to have more than one step within a cycle statement if the increments are not equal. For example:

```
CYCLE/DEEP,DEPTH,70,STEP,30,STEP,15,MMPM,100
```

would be handled as expected by:

```
CYCLE/DEEP,DEPTH,#fedto,STEP,#step,MMPM,#fedupm
```

In this example the first feed would be to 30mm, then 45mm and 60mm, and finally to 70mm. It is not necessary to place "STEP" or "**#step**" in the template more than once, and any number of "STEP,value" pairs could appear in the cycle statement.

steplist—A list of incremental depths for each successive feed. This differs from the prior example in that the values are not separated by repetitive minor words. For example:

```
CYCLE/DEEP,DEPTH,70,STEP,20,15,10,MMPM,100
```

would be represented by:

```
CYCLE/DEEP,DEPTH,#fedto,STEP,#steplist,MMPM,#fedupm
```

In this case successive feeds would be to 20mm, 35mm, 45mm, 55mm, 65mm and 70mm.

steprap—Distance above the prior depth that the tool will move to before the next deeper feed. If the cycle retracts completely between successive feeds, this motion is rapid and in the downward direction after each retract. If the cycle only partially retracts between feeds, then this motion is in the upward direction and may or may not be rapid.

tlangl—The included angle of the tool tip for countersink operations. Only meaningful if a **cskdia** is also present.

Modal specification

Each cycle definition can specify a number of modal values to control details of the cycle motion. All modals have default values, and it is not necessary to define the settings of those that take these default conditions. If all of a definition's modals take the defaults, the "MODAL" line can be omitted.

Following is an explanation of each modal and its default value, in alphabetic order:

auto_mode—(default OFF) Controls whether the automatic retract plane should be used. When set to "ON", any value corresponding to the **rtreto** mnemonic will be ignored and the final retract of the cycle will be to the automatic retract plane. Cycle statements may differ by only "AUTO" as a minor word. Rather than requiring two similar cycle templates, VERICUT will turn this modal "ON" if a cycle statement contains "AUTO". It is thus only necessary to include this modal in the "MODAL" line of a cycle definition, if some other feature of the cycle statement is the trigger.

cycle_op—(default ON) Controls whether cycle statements that match a definition are to be expanded at all. An alternative to "cycle_op=OFF" would be to toss out the entire definition, but then the VERICUT log would report unrecognized cycles. Use this modal if you wish emulate some cycles but not others.

deep_retract—(default ON) Controls whether the tool should retract completely between successive feeds. When turned "ON", the tool will retract to the height corresponding to the **rtcto** mnemonic (or **rapto** if there is no **rtcto**) between feeds. If "deep_retract=OFF", then between feeds the tool will retract incrementally by the distance corresponding to the **steprap** mnemonic, or simply dwell if there is no such value. This modal has no meaning if there is only one feed depth.

retract_mode—(default RAPID) Controls the feed rate used for all retract motions in the cycle. With the default setting, all retracts will be rapid. Other options are "retract_mode=FEDRAT" and "retract_mode=NONE". "FEDRAT" would be used when the retract motion should be at the same speed as the feed, such as in a ream or tap cycle. "NONE" would be used to suppress retract motion completely, as in a milling cycle.

thru_mode—(default OFF) Controls the feed rate of successive feeds. When this modal is "ON", the first feed is at the feed rate, the second is rapid, the third is at the feed rate, etc. This can be used when the tool is passing through several webs, and it can be moved quickly through the air between the webs. This modal has no meaning if there is only one feed depth.

Applicability (USEWITH)

The USEWITH line of a cycle definition can list the names of one or more flavors of APT. All of the unique flavors present in a user file are presented in a choice box on the "**Cycles**" tab of the **Setup menu > APT Settings** window. When you select one of the flavors in the choice box, VERICUT will only consider corresponding cycle definitions when processing your APT tool paths.

If you add a new cycle set, for a different flavor of APT or a different post-processor, you can give it any name you wish, provided the name does not contain spaces or other APT delimiter. So your cycle definitions may each end with:

USEWITH:MYPOST

An alternative would be to delete all the cycle definitions in the user file and insert yours, omitting the USEWITH line for each. Then the choice box would just offer "All", however, only your definitions would be present.

Cycle definition sets

In alphabetic order, the sets of cycle definitions provided in the default user files simulate the following flavors of APT-CLS canned cycles:

ACL

CADRA

CATIA

CONVERTED—generated by VERICUT via reverse post-processing from G-code

CV—ComputerVision

Help on VERICUT

GEN4AX—a post used with NCL

NCL—combines GEN4AX and POSTWORKS sets

PTC—Pro/Manufacturing

POSTWORKS—a post used with NCL

UG— Unigraphics CLS

Simulating G-code tool paths

Setting the tool path programming and scanning methods

After opening a Machine file and Control file, the next most important piece of information for VERICUT to interpret a G-code tool path is setting the **Programming Method**. Depending on the G-code data, you may also need VERICUT to perform a special "scan pass" to identify jump destinations for looping-branching logic, or to ignore specific columns of data. (See below)

To set the programming method of the G-code tool path(s):

1. Click **Setup menu > G-code > Settings**.
2. Click the **Settings** tab.
3. Choose the **Programming Method** that describes the relationship between cutting tools and what is driven by tool path file motion commands.
 - Choose **Tool Tip** to drive the center of the tool tip.
 - Choose **Gage Length** to drive a gage length reference point, such as spindle face or rotary head pivot point.
 - Choose **Tool Length Compensation** to drive the tool tip in tool length comp. mode (e.g. G43) and drive the gage length reference point (see above) when tool length comp. is off.
4. Press **OK** or **Apply**.
5. If you selected **Gage Length** or **Tool Length Compensation** programming methods, supply tool gage offsets via tool properties or entries in a Gage Offset table.
6. Although VERICUT automatically calculates length compensation offsets based on the Input Program Zero table location and gage offsets of each tool, you may manually enter values in a Tool Length Compensation table if you wish.

To set the scanning method for the G-code tool path(s):

1. If tool path(s) being processed contain looping-branching logic, such as: subroutines, NC macros, DO/WHILE loops, IF/THEN checks, GOTO jumps, etc., select **Scan Toolpath Files** to enable VERICUT to determine branch locations that will be needed during processing.

Note: If your tool path does not contain looping-branching logic, performing a scan pass causes an unnecessary delay in processing time. Clear the **Scan Toolpath Files** checkbox for processing these simple tool path files.
2. If valid G-code data (M and G-codes, axis commands, etc.) resides in a fixed number of columns, select **Ignore Data after Column** and enter the last valid G-code data column number. This will prevent VERICUT from attempting to process comments and other characters that can reside in the columns that follow.
3. Press **OK** or **Apply**.

Adding, modifying, or deleting variables

VERICUT can support variable handling when a special word is defined as a "Variable Tag". For example, for "#1" to be recognized as a variable, the "#" word must be defined with **Special Sub-type=Variable Tag**. Similar consideration must be given to the variable assignment character and other math characters, if used in the tool path file. For example, for the block "#1=100" to store a value of "100" with variable "1", the "=" word must be defined with **Math Sub-type=Assignment**. Ref. **Setup menu > Control > Word Format** for more information.

Variables are assumed to have zero for a default value. Use the procedures below to initialize variables with a specified value, as well as modify or delete existing variables.

Note: Variable setting is not necessary when tool path data blocks set variable values, e.g. G10L2P...

To add a new variable:

1. Click **Setup menu > G-code > Settings**.
2. Click the **Variables** tab.
3. Enter the variable **Name**. For example, the name for variable #100 is "100".
4. Choose the variable **Type** which indicates the type of data to be assigned.
5. Enter the **Value** to initialize the variable with.
6. Click **Add**.

The new variable is added, and appears in the Variables window.

To modify an existing variable:

1. Click **Setup menu > G-code > Settings**.
2. Click the **Variables** tab.
3. In the list of variables, select the variable to modify.
4. Modify variable data as required.
5. Click **Modify**.

The modified variable appears in the Variables window.

To delete variables:

1. Click **Setup menu > G-code > Settings**.
2. Click the **Variables** tab.
3. In the list of variables, select the variable to delete.
4. Click **Delete** to delete the selected variable, or **Delete All** to delete all variables.
5. Click **Yes** to confirm deletion.

Help on VERICUT

The deleted variable(s) are erased from the Variables window.

Adding, modifying, or deleting tables or table rows

Tables are used to store values for control features, axes values for important machine locations, and other data related to processing G-code tool path files.

The following VERICUT windows provide access to tables:

Setup menu > G-Code > Settings: Job Tables tab—Specify work offsets and machine locations specific to the current tool path file(s).

Setup menu > G-Code > Settings: Tool Tables tab—Store tool-related offset and compensation data.

Setup menu > Machine > Settings: Tables tab—Specify important machine locations and detail tool change motion.

Note: The following procedures assume you already selected one of the tabs on the windows listed above, based on your table needs.

To add a new table or table row:

1. Choose the name of the desired table from the **Tables list**, or the **Table Name** option list. (See list of available tables)
2. If necessary, choose the **SubSystem ID** for which the table data is being defined.
3. Enter the **Index** or **Register** value, and corresponding **Value(s)** to store in the table. Separate multiple-value entries by spaces.

For table-specific information, such as data entry formats and their affect in VERICUT, refer to "Tables - listed alphabetically" in the *Reference* chapter.

4. Click **Add**.

The new table/row is added, and appears in the Tables window.

To modify an existing table row:

1. Choose the name of the table to modify from the **Tables list**, or the **Table Name** option list.
2. Select the table row to modify.
3. Modify table data as required.
4. Click **Modify**.

The modified table row appears in the Tables window.

To delete a row from a table:

1. Choose the name of the table containing the row to delete (choose from the **Tables list**, or the **Table Name** option list).
2. Select the table row to delete.
3. Click **Delete** to delete the selected row.

Help on VERICUT

4. Click **Yes** to confirm deletion.

The deleted row is erased from the Tables window. If the last table row is delete, the table is also erased.

Adding, modifying, or deleting subroutines

Subroutines, or "subs" for short, contain additional programming or machine commands, and can exist within a G-code tool path file or reside in files separate from the main G-code program. To simulate data contained in external subroutine files, use the procedures below to make VERICUT aware of the file names and locations.

The following VERICUT windows provide access to external subroutine files:

Setup menu > Control > Adv. Options: Subroutines tab—Lists "control subroutine" file names.

Setup menu > G-Code > Settings: Subroutines tab—Lists "job subroutine" file names.

Note: The following procedures assume you already selected one of the tabs on the windows listed above, based on how you want VERICUT to handle the subroutine data. (Ref. "Types of external subroutines" in this chapter.)

To add a subroutine file name:

1. Click **Add**.
2. Type or select the **File name** of the subroutine file, then click **Open**.

The new subroutine file name is added, and appears in the subroutine files list.

To change a name in the subroutine file list to point to a different file:

1. Choose the name of the sub file in the subroutine files list.
2. Click **Rename**.
3. Type or select the replacement sub **File name**, then click **Open**.

The renamed sub file appears in the subroutine files list. This is really just a short cut to deleting, the adding a new sub file name.

To delete a subroutine from the files list:

1. Choose the name of the sub file to delete from the subroutine files list.
2. Click **Delete** to delete the selected sub, or click **Clear** to delete all the subs from the list.

Deleted subs are erased from subroutine files list. Note that deleting a sub file name from the VERICUT configuration does not delete the file from your computer.

Controlling message output destinations

Use the procedure below to set the destination and format of messages output during a simulation.

To set output message destination and format:

1. Click **Setup menu > G-Code > Process Options > Messages**.
2. Choose the desired location(s) for each message type. (See window features)
3. Choose the **Output Filenames Format** for messages listing files used by VERICUT.
4. Set **Output "Word is not defined" Errors** accordingly, depending on if you want to see these messages when G-code/addresses are encountered that the control is not set up to handle.
5. Press **OK**.

Configuring debug message output

Use the procedure below to set debug messages to output, and where to send them.

To output debug messages during processing:

1. Click **Setup menu > G-Code > Process Options > Debug**.
2. Configure the Debug window as required. (See window features)
3. Press **OK**.

Note: The output destination for debug messages is controlled by **Setup menu > G-Code > Process Options > Messages: Send Debug Messages**.

Configuring "Block Skip" G-code settings

Use the procedure below to specify the block skip character, and control which levels of block skipping are recognized.

To specify the block skip character and control block skipping:

1. Click **Setup menu > G-code > Settings**.
2. Click the **Block Skip** tab.
3. Enter the **Block Skip Character**.
4. Set the **Apply Switch Value** option for when to reference block skip switch values:
 - **Immediately**—Referenced when VERICUT is reset, upon loading a new User file, and any time tool path processing is started.
 - **At Reset**—Referenced when VERICUT is reset, or upon loading a new User file.
5. By default, all levels of block skipping are turned on. If desired, clear "Switch" checkboxes of block skip levels to ignore.
6. Press **OK** or **Apply**.

Configuring "Wire EDM" G-code settings

Use the procedure below to configure settings for wire EDM machining.

To configure wire EDM settings in the control:

1. Click **Setup menu > G-code > Settings**.
2. Click the **WireEDM** tab.
3. Configure the window as required. (See window features)
4. Press **OK** or **Apply**.

See also: Configure Wire EDM control settings

Reverse post-processing G-codes to APT

Use the procedure below to convert, or "reverse post-process" G-code data into ASCII APT tool path records.

To convert (reverse post-process) G-codes to an ASCII APT tool path:

1. Assuming VERICUT is configured to process a G-code tool path, click **Setup menu > G-Code > Process Options > Output Filenames**.
2. Type or select the **APT Output File** to receive converted tool path records.
3. If desired, type or select the **G-code Log File** to receive error, warning and informational messages about G-code processing. (Recommended during development of reverse post-processors.)
4. Ensure **Create APT Output File** is selected.
5. Press **OK**.
6. Reset VERICUT (press ) to ensure changes are effective, then start the simulation.

Conversion occurs during tool path processing. After cutting, you can view the APT Output file via **Setup menu > G-Code > Process Options > View APT Output File**, and view the G-Code Log file via **Setup menu > G-Code > Process Options > View G-Code Log File**.

Viewing the converted APT Output file

After reverse post-processing, use the procedure below to view the output file containing the converted ASCII APT tool path records.

To view the APT Output file resulting from reverse post-processing:

Click **Setup menu > G-Code > Process Options > View APT Output File.**

Creating and viewing a G-code Log file

The G-Code Log file contains error, warning and informational messages about G-code processing. One of the most valuable pieces of information this file provides is a copy of each input G-code data block followed by the converted ASCII APT tool path record indented by 4 spaces. This allows you to see a direct translation of G-code versus equivalent ASCII APT output for all tool path records processed.

Use the procedures below to create and view a G-Code Log file. **Note:** Additional considerations apply for reverse post-processing G-codes to APT, see: Reverse post-process G-codes to APT for steps that accomplish this task.

To create a G-Code Log file:

1. Assuming VERICUT is configured to process a G-code tool path, click **Setup menu > G-Code > Process Options > Output Files**.
2. Type or select the **G-code Log File** to receive error, warning and informational messages about G-code processing. (Recommended during development of reverse post-processors.)
3. Press **OK**.
4. Reset VERICUT (press ) to ensure changes are effective, then start the simulation.

To view the G-Code Log file:

Click **Setup menu > G-Code > Process Options > View G-Code Log File**.

Describing cutting tools

About describing cutting tools for VERICUT

VERICUT requires descriptions of the cutting tools used to machine the workpiece to accurately simulate the material removal process. VERICUT can process all standard APT CUTTER records and UG TLDATA records to obtain descriptions of the cutting tools, or you can define cutting tools via VERICUT's Tool Manager and store them in a Tool Library file. When non-cutting tool holders are defined and used in the simulation, VERICUT detects collisions that would occur with tool holder(s) and the workpiece or holding fixtures.

If VERICUT does not have a description of the cutting tool when tool path processing begins, the previous tool is used. If a previous tool has not been defined, an error similar to "Tool not defined - motion ignored" is issued.

A summary of the cutting tools used during the simulation and machine cycle time elapsed for each tool is written to the Log file.

VERICUT uses two types of cutting tools, depending on the machining type that will be using them. Descriptions follow.

Milling tools

Turning tools

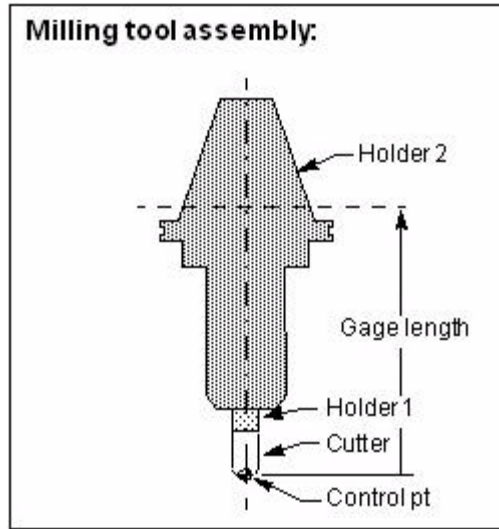
Milling tools

A milling tool assembly typically consists of three components:

Cutter—Cutting portion of the tool (known in VERICUT as "Cutter")

Shank—Non-cutting portion of the tool attached to the cutter (known in VERICUT as "Holder 1")

Holder—Grips the cutter/shank and holds the tool in the NC machine (known in VERICUT as "Holder 2")



Additional components can also be found in milling tool assemblies, such as: chucks, extensions, adapters, etc. For simulation purposes, these can be incorporated into one of the "Holder" components available in VERICUT.

Milling tool control point

The tool control point, or "driven point" is a point relative to the tool assembly that the NC data is commanding to move. By default, VERICUT assumes the mill tool tip (bottom center of the tool) is being driven. However, this relationship can be changed via the **Control Point** feature on the **Setup menu > Motion** window, or by including a **Control Point Offset** in the Tool Manager-defined tool properties.

In cases where a gage length reference point is being driven, such as with G-code data for a machine expecting to drive the spindle face or rotary pivot point, a gage length or "gage offset" value is often used. This value establishes the expected length for the tool to be extended from the holder. VERICUT realistically simulates this programming method by allowing you to enter these values in a Gage Offset table, or by including a **Gage Length** in the Tool Manager-defined tool properties.

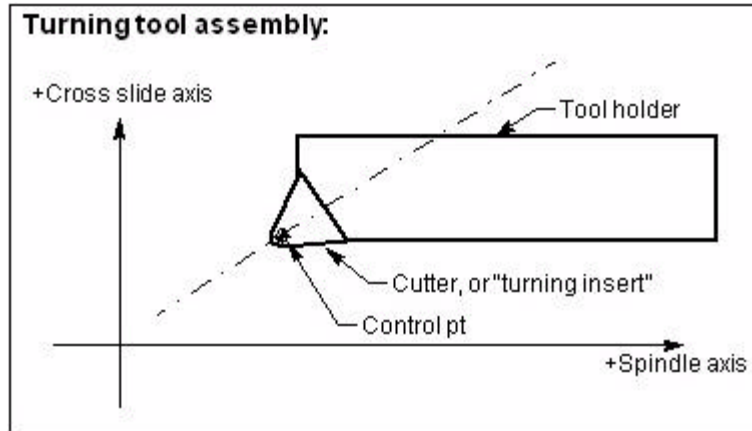
Simulated material removal is based on motion commands and the tool shape, relative to the control point.

Turning tools

A turning tool assembly typically consists of two components:

Cutter—Cutting portion of the tool sometimes referred to as an "insert" (known in VERICUT as "Cutter")

Holder—Grips the cutter and holds the tool in the NC machine (known in VERICUT as "Holder 1")



Additional components can also be found in turning tool assemblies, such as: chucks, extensions, adapters, etc. For simulation purposes, these can be incorporated into one of the "Holder" components available in VERICUT.

Turning tool control point

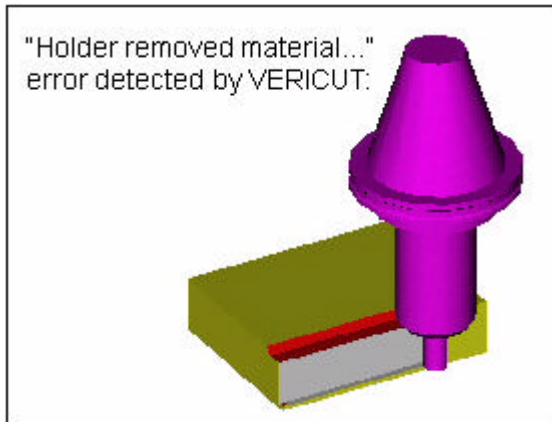
The tool control point, or "driven point" is a point relative to the tool assembly that the NC data is commanding to move. By default, VERICUT assumes the center the turning tool tip is being driven. However, this relationship can be changed by including a **Control Point Offset** in the Tool Manager-defined tool properties.

In cases where a gage length reference point is being driven, such as with G-code data for a machine expecting to drive the turret face or center, gage offset values are often used. These values establish where the NC data control point is, relative to the actual machine driven point. VERICUT realistically simulates this programming method by allowing you to enter these values in a Gage Offset table, or by including a **Gage Offsets** in the Tool Manager-defined tool properties.

Simulated material removal is based on motion commands and the tool shape, relative to the control point.

Defining profile tool shapes

Using profile tool shapes, also known as "special cutters" extends the ability to describe tool shapes to include cutters having almost any cutting shape and differentiate the non cutting portions of the tool assembly, such as: shanks, extensions and holders. Material removed by the non-cutting portion of a tool assembly is immediately flagged as an error, and shaded using the red **Error** color.



Tools having special cutter profiles are typically defined via the Tool Manager using "Profile" tool shapes. However, they can also be described using special tool path record formats (see below).

CUTTER record special cutter profile formats:

CUTTER/XCUT, PT, ...profile description

CUTTER/SHANK, PT, ...profile description

CUTTER/HOLDER, PT, ...profile description

PPRINT/VERICUT-TC record special cutter profile formats:

PPRINT/VERICUT-TC XCUT, PT, ...profile description

PPRINT/VERICUT-TC SHANK, PT, ...profile description

PPRINT/VERICUT-TC HOLDER, PT, ...profile description

G-code version of PPRINT/VERICUT-TC record special cutter profile formats (Fanuc control example):

(VERICUT-TC XCUT, PT, ...profile description)

(VERICUT-TC SHANK, PT, ...profile description)

(VERICUT-TC HOLDER, PT, ...profile description)

Special cutter profiles are defined differently, depending on the tool type: for mill or turn. Consult the following sections for specific information.

Milling tool profiles

Turning tool profiles

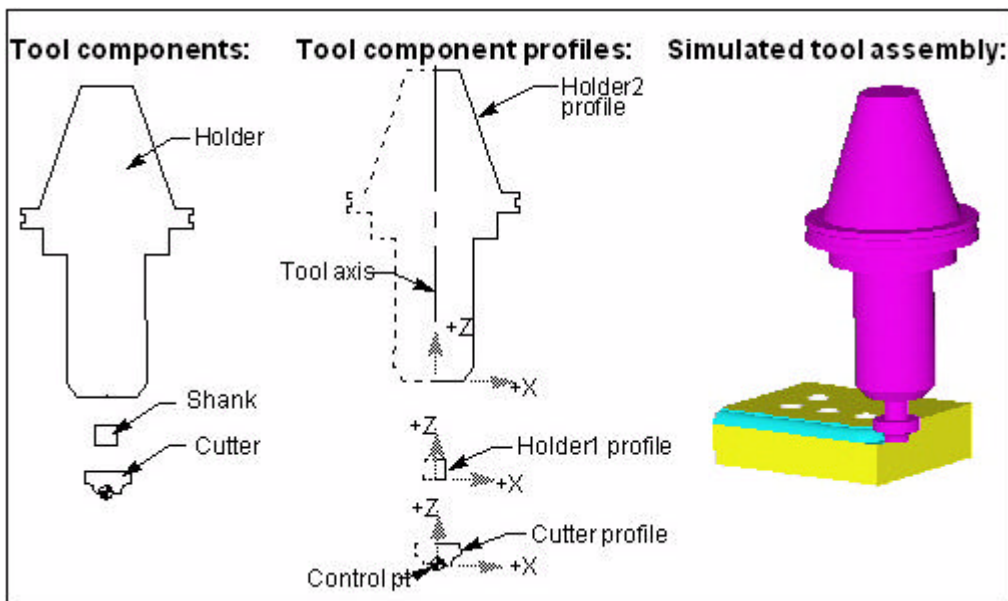
Milling tool profiles

Milling tool profiles are defined using a 2-D profile composed of line and arc segments in the ZX plane. The profile of each tool component (cutter or holder) is described in its own local coordinate system where the Z-axis is the tool centerline.

The tool components “stack up”, providing the ability to mix and match cutter and holder components. The holder origin is placed on top of the cutter shape. If two holders are defined, "Holder 2" stacks on "Holder 1" (formerly known as a "shank"), which stacks on top of the cutter. More will be discussed on this topic later in this section. Holders can also be stacked on top of standard cutters defined in the tool path file.

By default, VERICUT assumes the origin of the cutter component is being driven. However, this relationship can be changed via the **Control Point** feature on the **Setup menu > Motion** window, or by including a **Control Point Offset** in the Tool Manager-defined tool properties.

Sample tool assembly with milling tool profiles:



Milling tool profile data elements

The data elements used to define milling tool profiles are listed below. The profile for each tool component is defined relative to its own local coordinate system.

Point X,Z—Defines a point on the profile. When a point data element follows a previous point, a line segment is created connecting the points. When a point follows an arc, the point is used to limit the arc segment.

Points X,Z,X,Z,...X,Z—Used only with special tool path record formats, this data element defines a series of points on the profile, each connected with a line segment.

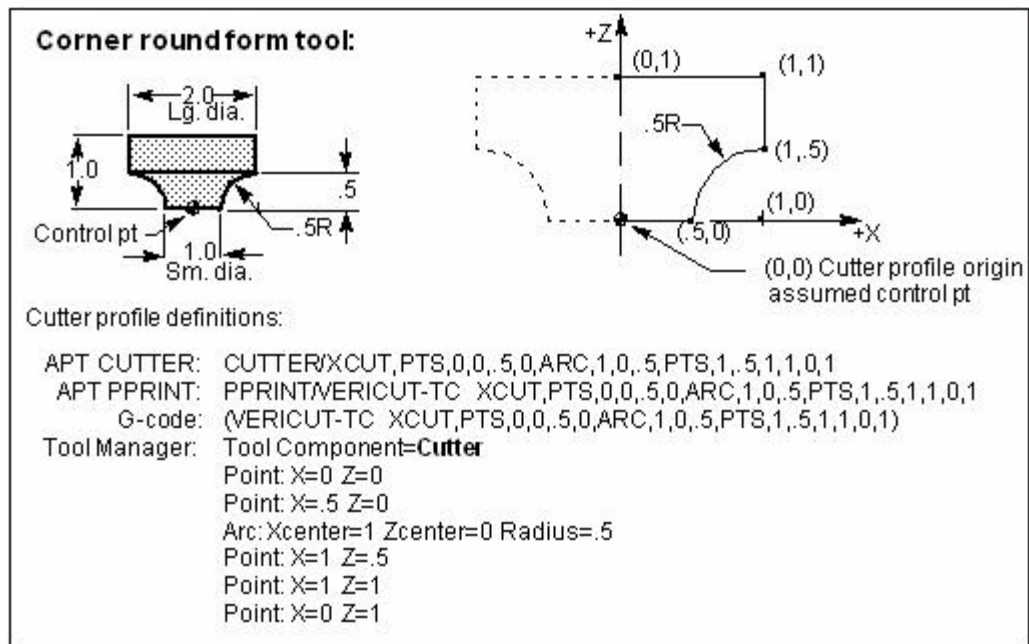
Arc Xcenter,Zcenter,Radius [CW, CCW, SHORTEST_DISTANCE]—Defines an arc segment on the profile. Control the Arc direction using the “CW”, “CCW”, or “Shortest distance” options.

An Arc must be limited by a preceding and trailing point. Tool profiles can not begin with, end with, or have two successive Arc data elements.

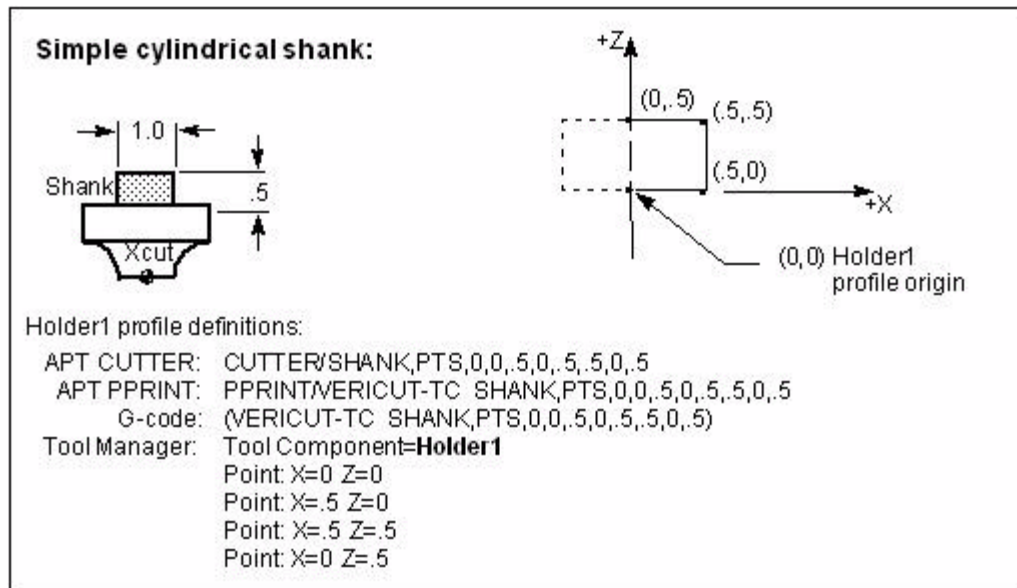
Rules for defining milling tool profiles

1. The profile definition must begin with a Point at X=0, and may begin anywhere on the Z-axis (tool axis). Recall that X0Z0 of a cutter profile is assumed by VERICUT as the control point.
2. The profile must have at least three points.
3. In general, the profile definition is continues in a counter-clockwise direction, and ends with a point that is higher in Z than the beginning point. VERICUT automatically closes any profile that does not end on the Z-axis by supplying an ending point at X=0 and perpendicular to the tool axis.
4. The profile can not have any portion be zero diameter or self-intersecting (cross over on itself).
5. Any tool component profile is permitted to turn once in the negative Z direction (examples follow).

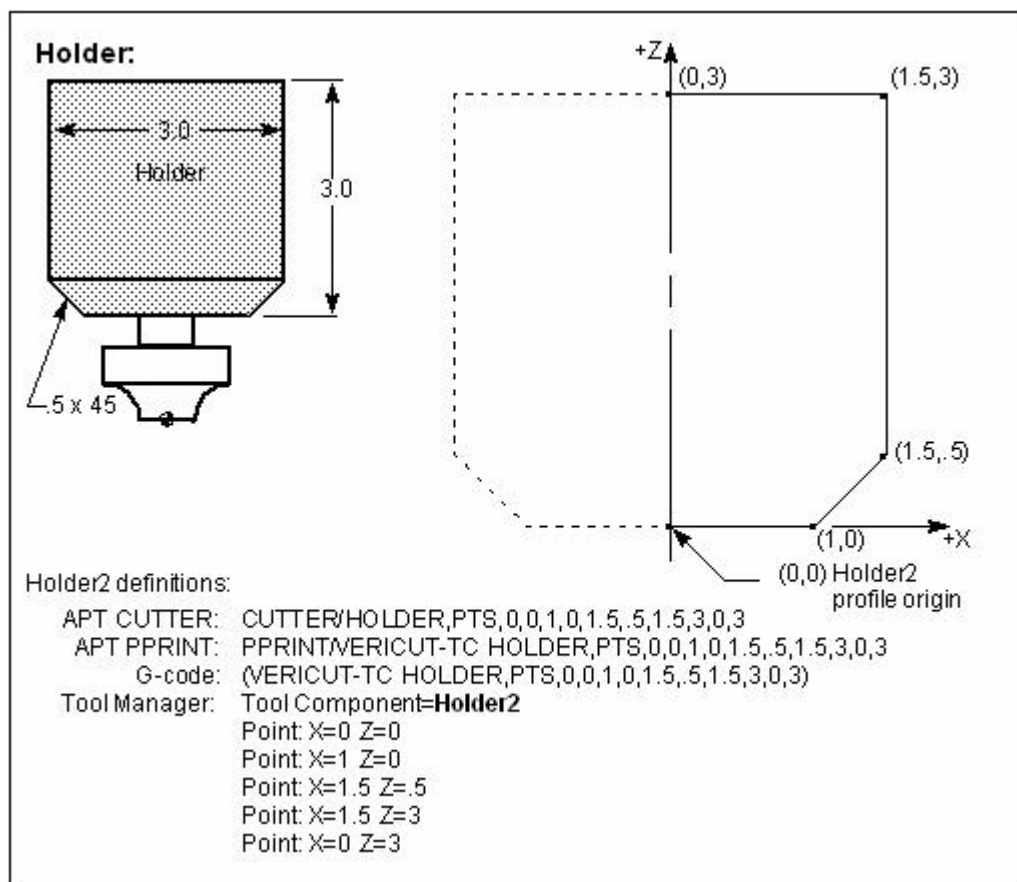
Example defining a milling cutter profile:



Example defining a shank, or "Holder1" profile:



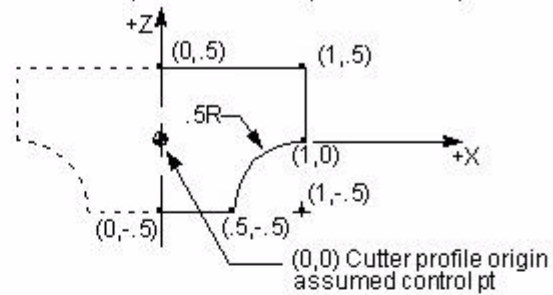
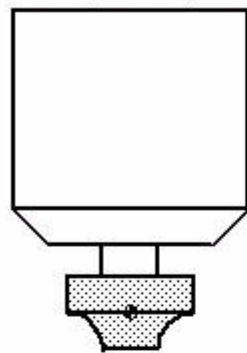
Example defining a tool holder, or "Holder2" profile:



Examples of using "Negative Z" profiles:

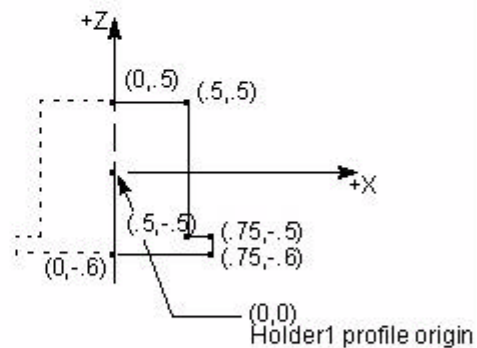
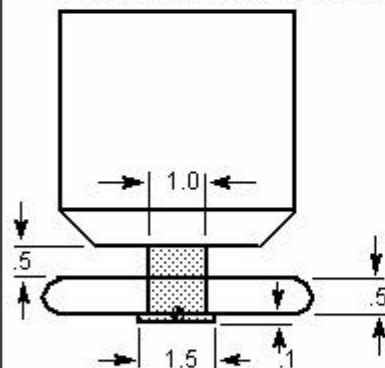
Example 1:

Starting a cutter profile with -Z to alter the control point-to-tool shape relationship.



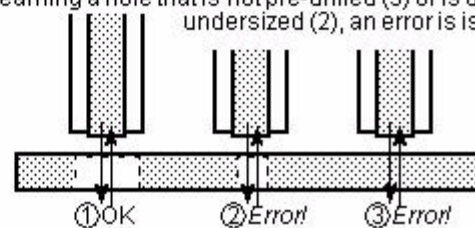
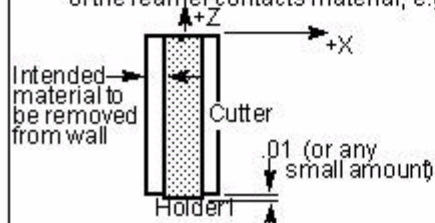
Example 2:

Starting Holder1 profile with -Z to define a non-cutting portion of the tool below the cutter (e.g. a retaining nut under a wheel cutter).



Example 3:

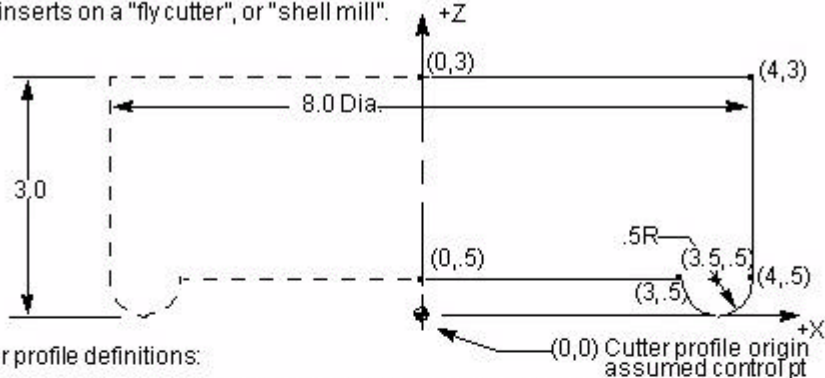
Starting Holder1 with -Z to define the non-cutting bottom of a reamer cutter. If the bottom of the reamer contacts material, e.g. reaming a hole that is not pre-drilled (3) or is drilled undersized (2), an error is issued.



Example cutter profile with a “Negative Z” turn

Example 4: "fly cutter" or "shell mill" with .5R inserts

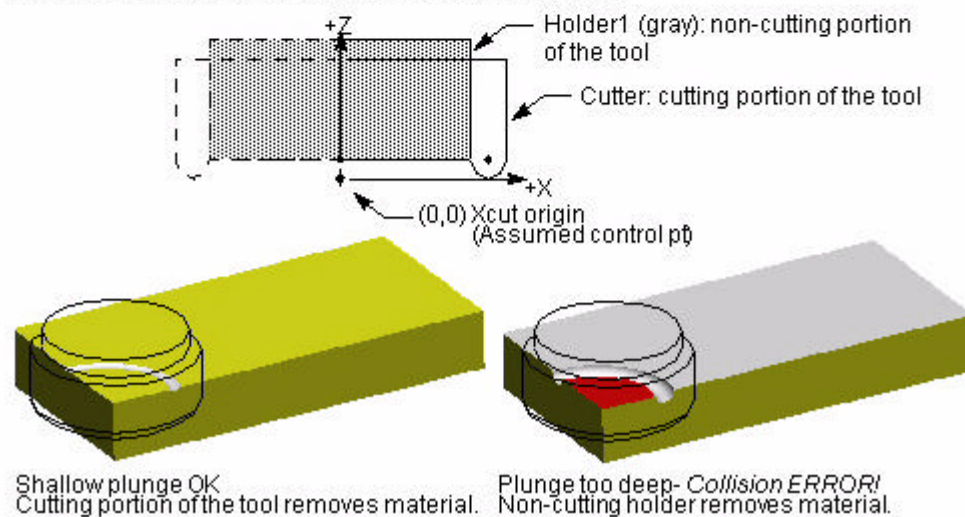
Defining a cutter profile with a turn in negative Z to simulate inserts on a "fly cutter", or "shell mill".



Cutter profile definitions:

```
APT CUTTER: CUTTER/XCUT,PTS,0,.5,3,.5,ARC,3.5,.5,.5,PTS,4,.5,4,3,0,3
APT PPRINT: PPRINT/VERICUT-TC XCUT,PTS,0,.5,3,.5,ARC,3.5,.5,.5,PTS,4,.5,4,3,0,3
G-code: (VERICUT-TC XCUT,PTS,0,.5,3,.5,ARC,3.5,.5,.5,PTS,4,.5,4,3,0,3)
Tool Manager: Point: X=0 Z=.5
               Point: X=3 Z=.5
               Arc: Xcenter=3.5 Zcenter=.5 Radius=.5
               Point: X=4 Z=.5
               Point: X=0 Z=3
```

Add Holder1 for "plunge collision detection", for example:

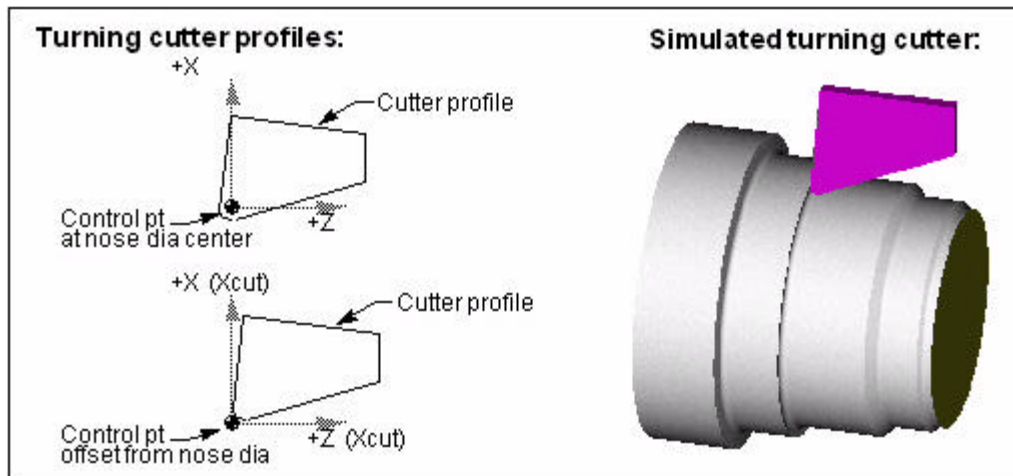


Turning tool profiles

Turn tool profiles are used only to describe the cutter tool component, and are defined using a "closed" 2-D profile composed of line and arc segments in the ZX plane. The profile is described in a local coordinate system where the origin of the cutter component is assumed as the tool control point.

By default, VERICUT assumes the origin of the cutter component is being driven. However, this relationship can be changed via including a **Control Point Offset** in the Tool Manager-defined tool properties.

Sample turning tool profile:



Turning tool profile data elements

The data elements used to define turning tool profiles are listed below. The profile is defined relative to its local coordinate system. Note that Z=spindle axis and X=cross-slide axis.

Point Z,X—Defines a point on the profile. When a point data element follows a previous point, a line segment is created connecting the points. When a point follows an arc, the point is used to limit the arc segment.

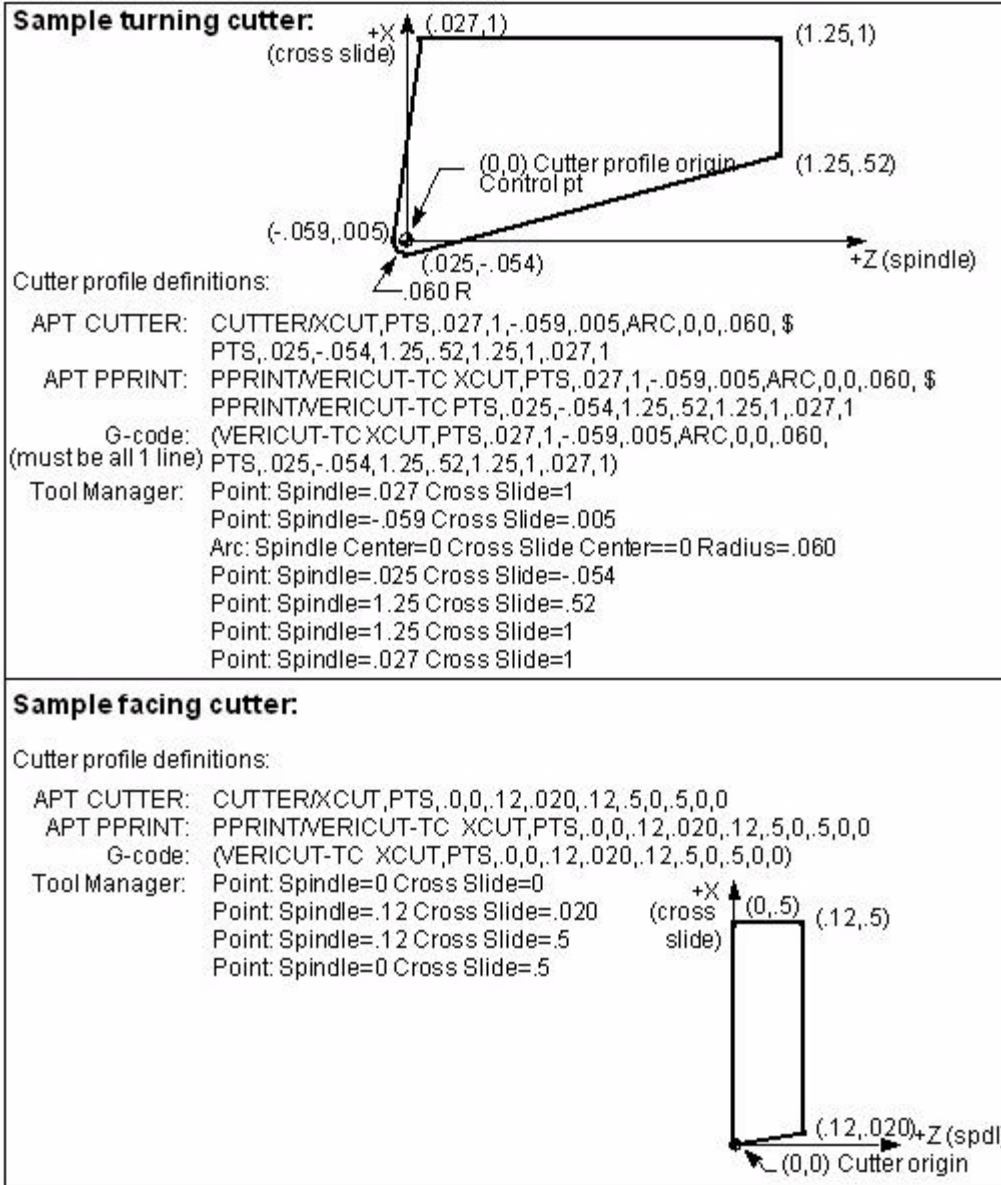
Points Z,X,Z,X,...Z,X—Used only with special tool path record formats, this data element defines a series of points on the profile, each connected with a line segment.

Arc Zcenter,Xcenter,Radius [CW, CCW, SHORTEST_DISTANCE]—Defines an arc segment on the profile. Control the Arc direction using the “CW”, “CCW”, or “Shortest distance” options. An Arc must be limited by a preceding and trailing point. Tool profiles can not begin with, end with, or have two successive Arc data elements.

Rules for defining turning tool profiles

1. The profile definition must begin with a Point, and may begin anywhere in X and Z. Recall that X0Z0 of a cutter profile is assumed by VERICUT as the control point.
2. The profile must have at least three points.
3. In general, the profile definition is continues in a counter-clockwise direction, and ends with a point at the same location as the beginning point (creates a closed profile shape). VERICUT automatically closes any profile that does not end on the beginning point by supplying an ending point at that location.
4. The profile can not be self-intersecting (cross over on itself).
5. A profile is permitted to turn once in the clockwise direction to create a concavity, such as an interior angle greater than 180 degrees or concave arc.

Examples defining turning cutter profiles:



Using the Tool Manager

Creating a new Tool Library

Tool Library files contain descriptions of cutting tools, or tool assemblies. Tool Library data is used by VERICUT when the **Setup menu > Toolpath** function is configured to do so. Use the procedure below to create a new Tool Library.

To create a new (blank) Tool Library file:

1. Click  on the Toolbar, or click **Setup menu > Tool Manager**.
2. From the Tool Manager window, click **File menu > New**.

All data is cleared from the Tool Manager.
3. Press **Add** to add new tools. (See how)
4. To save the new Tool Library file, click **File menu > Save As**.
5. In the window that opens, select or type the name of the file to save and click **Save**.

Saving or opening a Tool Library

Tool Library files contain descriptions of cutting tools, or tool assemblies. Tool Library data is used by VERICUT when the **Setup menu > Toolpath** function is configured to do so. Use the procedure below to create a new Tool Library.

To save a new Tool Library file:

1. Assuming you are in the Tool Manager window, click **File menu > Save As**.
2. In the window that opens, select or type the name of the file to save and click **Save**.

To open an existing Tool Library file:

1. Click  on the toolbar, or click **Setup menu > Tool Manager**.
2. From the Tool Manager window, click **File menu > Open**.
3. In the window that opens, select or type the name of the file to open and click **Open**.

The Tool Library file is opened. The first tool in the library is selected and displayed in the Tool Manager window.

Tips:

If errors occur during opening, consult the *ERRORS and Problem Solving* chapter for problem descriptions and recommended solutions.

Adding a tool to the library

Tool Library files contain descriptions of cutting tools, or tool assemblies. Tool Library data is used by VERICUT when the **Setup menu > Toolpath** function is configured to do so. Use the procedures below to add new tools to a Tool Library. (See how to create a new library)

For ease of understanding, this section is divided into the basic tasks below. Consult to appropriate task to describe specific tool components, or follow them in order to describe a tool assembly. All tasks assume the

Tool Manager window is already open (click  on the toolbar, or click **Setup menu > Tool Manager**).

Describe the tool ID, description, and how it will be used

Describe a cutter

Describe a milling tool shank

Describe a tool holder

Specify tool properties

Add the current tool description to the library

To specify the tool ID, description, and how it will be used in VERICUT:

1. Assuming you are in the Tool Manager window, click **Add**.
2. Enter an **ID** and **Description** to identify the tool.
3. Choose the **Tool Type** that represents the type of machining the tool perform.
4. Set the **Units** in which to enter tool values.

Continue with describing a cutter, shank, or holder tool component.

To describe a cutter (cutting portion of the tool):

1. Click **Tool Component=Cutter**.
2. Press **Copy** to copy an existing cutter, or choose the icon having the desired shape and enter supporting data to describe the cutter. (See mill tool icons or turn tool icons and descriptions)

For information about defining **Profile** tool shapes, see "Defining profile tool shapes" in this chapter.

3. In the **Tool Assembly** area, press **Add** to add the cutter component to the current tool assembly.

Continue with describing a shank or holder component, or specify the tool properties.

To describe a milling tool shank (non-cutting portion of the tool attached to the cutter):

1. Click **Tool Component=Holder1**.
2. Press **Copy** to copy an existing shank, or choose the icon having the desired shape and enter

supporting data to describe the shank. (See mill tool icons or turn tool icons and descriptions)

For information about defining **Profile** tool shapes, see "Defining profile tool shapes" in this chapter.

3. In the **Tool Assembly** area, press **Add** to add the shank component to the current tool assembly-appears as "Holder1".

Continue with describing a holder component, or specify the tool properties.

To describe a tool holder (holds the tool in the NC machine):

1. Click **Tool Component=Holder2**.
2. Press **Copy** to copy an existing holder, or choose the icon having the desired shape and enter supporting data to describe the holder. (See mill tool icons or turn tool icons and descriptions)

For information about defining **Profile** tool shapes, see "Defining profile tool shapes" in this chapter.

3. In the **Tool Assembly** area, press **Add** to add the holder component to the current tool assembly-appears as "Holder2".

Continue with specifying the tool properties.

To specify tool properties:

1. Click **Properties**.
2. In the window that opens, specify the properties of the tool. (See window features)
3. Click **OK**.

Continue with adding the current tool description to the library.

To add the current tool description to the library:

1. Examine the current tool assembly displayed in the Tool Manager window **Tool Display** area to verify that it is correct.
2. If correct, click **OK** or **Apply** on the Tool Add window to add the current tool description to the library.

Modifying, copying, or deleting a tool in the library

Use the procedures below to modify, copy, or delete tools in a Tool Library file. All tasks assume the Tool Manager window is already open (ref. **Setup menu > Tool Manager**).

To modify an existing tool in the library:

1. Assuming you are in the Tool Manager window, highlight the tool to modify in the Tool/ID list.
2. Click **Modify**.
3. Modify the tool values as required. (See window features)
 - In the **Tool Assembly** area, select the shape in the tool component to modify.
 - Modify tool shape values as needed. (See mill tool icons or turn tool icons and descriptions)
 - In the **Tool Assembly** area, press **Modify** to update the current tool assembly with the modified shape.
 - Repeat as necessary for other tool component shapes.
4. Examine the current tool assembly displayed in the Tool Manager window **Tool Display** area to verify that it is correct.
5. If correct, click **OK** or **Apply** to add the current tool description to the library.

To copy an existing tool in the library:

1. Assuming you are in the Tool Manager window, highlight the tool to copy in the Tool/ID list.
2. Click **Copy**.

The window opened is pre-configured with values copied from the selected tool.
3. Enter a unique tool **ID** and **Description**.
4. Modify window values as required. (See window features)
5. Examine the current tool assembly displayed in the Tool Manager window **Tool Display** area to verify that it is correct.
6. If correct, click **OK** or **Apply** to add the current tool description to the library.

To delete an existing tool from the library:

1. Assuming you are in the Tool Manager window, highlight the tool to delete in the Tool/ID list.
2. Click **Delete**.

The tool is deleted. If you deleted a tool by mistake, close the Tool Manager without saving the Tool Library file, then reopen for editing.

Controlling and monitoring the simulation

Ways to monitor a VERICUT simulation

This section introduces you to the many tools that VERICUT has for monitoring the simulation. Most functions that provide general information about your session are located in the **Info menu**.

What do you want to do?

- Learn about the "BUSY" and "OptiPath" lights

- See tool path files being processed

- See status of simulation and machine conditions

- View Log and Results files containing information about the VERICUT session

- Summarize files in use by VERICUT, and create copies of them for electronic sharing

Setting start and stop conditions for the simulation

In addition to the simulation control icons (VCR buttons), VERICUT provides a number of features to start and stop the simulation. Becoming familiar with this features will enable you to efficiently control the simulation.

To start at a specified line number:

1. Click **Setup menu > Motion**.
2. Enter the **Start At Line** number for the simulation to start at.
3. Press **OK** or **Apply**.
4. If VERICUT has already processed the line number to start at, press  to rewind the tool path to the beginning.
5. Start the simulation.

VERICUT fast forwards from its current location to the specified line while internally processing intermediate tool path records to store values for the current feed rate, spindle speed, coolant, tool description, and tool location. The simulation begins when the specified line number is reached.

To stop at a specified event (e.g. tool change, program stop, text, etc.):

1. Click **Setup menu > Motion**.
2. Choose the **Stop At** option for the desired event.
3. When applicable, enter supporting "stop at" text or values in the field next to this feature. (not case sensitive)

Example configuring to stop on "CLAMP CHANGE" text:



4. Press **OK** or **Apply**.
5. Start the simulation.

VERICUT stops the simulation when the event to stop at has occurred.

To stop after 1 or more errors or warnings:

1. Click **Setup menu > Motion**.
2. Select **Stop At Max Errors** (or warnings), then enter the quantity of errors to stop after.
3. Press **OK** or **Apply**.
4. Start the simulation.

Controlling animation speed

Using the **Setup menu > Motion** window features you can group cuts together for faster simulation, or slow the animation speed of material removal and/or simulated NC machine motions.

To adjust animation speed:

1. Click **Setup menu > Motion**.
2. Adjust the **Animation Speed** slider: slide left to slow, slide right to speed up.
3. Select **Control Machine Simulation Speed** to have **Animation Speed** also affect NC machine motions.
4. Press **Apply**.
5. Start the simulation and verify that animation speed is as desired. If necessary, readjust.
6. If sliding the **Animation Speed** slider all the way right is not fast enough, then enter a **Skip Cut** value indicating a quantity of motions to skip before updating the display.
7. Press **OK** or **Apply**.

Summarizing files in use by VERICUT

Use the procedure below to see a summary of the files and environment variables currently in use by VERICUT. A "**Copy all files**" option provides a quick and easy way to store a copy of the files in a new directory. This action is useful when preparing a set of files to send to CGTech, or other destination.

To see a summary of files in use:

Click **Info menu > File Summary**.

To copy all files in the summary to a directory:

1. Click **Info menu > File Summary**.
2. Click **Copy all files**  and choose a directory in which to store the copied files.

Tips:

You can easily configure VERICUT to use the copied files by running VERICUT from that directory, then opening the associated User file.

Analyzing models and machines

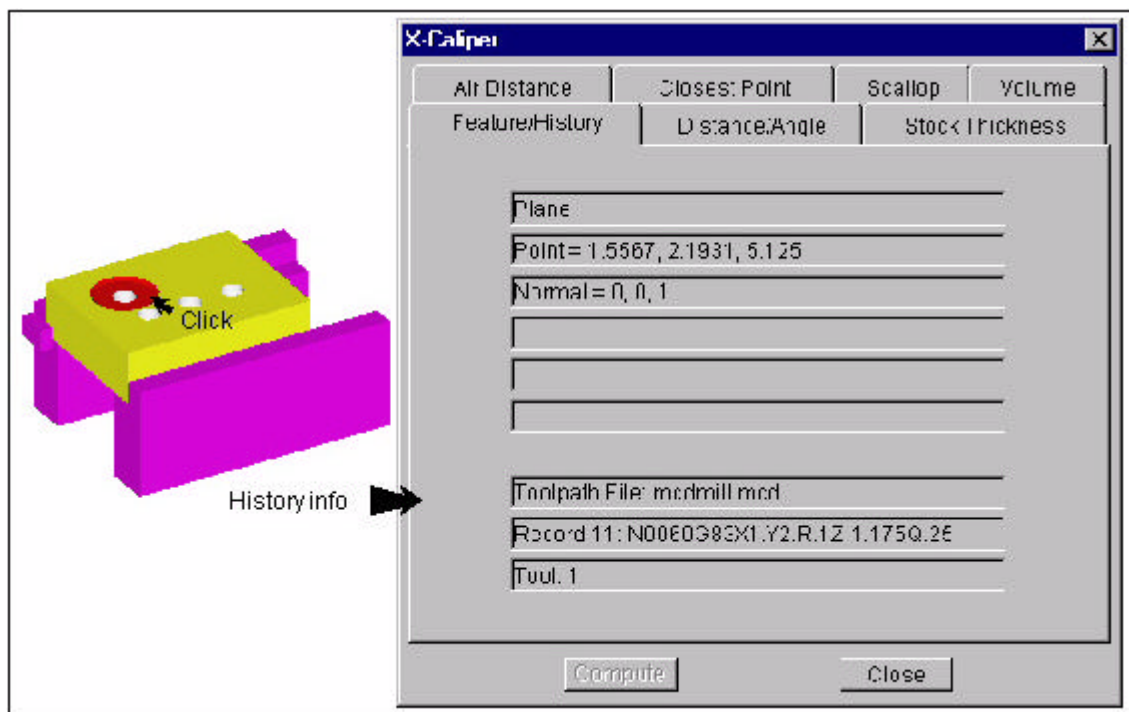
See feature or history info related to machine cuts

This **Feature/History** tab on the **Analysis menu > X-Caliper** window is commonly used to spot-check machined features, and determine the source of errors identified in red by VERICUT. The history data also informs you about which tool path records and tool path files were responsible for machining any feature on the workpiece.

To see feature or history info related to machine cuts:

1. Click  on the Toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Feature/History** tab.
3. Left-click on the cut in the workpiece to receive feature/history information on.

The feature/history information is displayed in the X-Caliper window.



Following are geometric features identified by VERICUT, listed alphabetically:

Auto-diff model—(X,Y,Z value and surface normal) an uncut surface on a model displayed after an auto-difference Solid method comparison.

Circle—(X,Y,Z center, axis vector and radius) a surface cut by a circular motion.

Cone—(X,Y,Z center, axis vector and side angle) a hole bottom cut by an angle tip drill, corner cut by a tapered endmill, or chamfer cut by a turning operation.

Cylinder—(X,Y,Z center, axis vector and radius) an inside corner cut by 2 linear motions having an interior angle less than 180 degree, fillet cut by a bull or ball endmill moving perpendicular to the tool axis, or diameter cut by a turning operation.

Ellipse—(X,Y,Z center, axis vector, major and minor focal distances) a ramping or plunging surface cut by the bottom of an endmill with a flat, or partially flat bottom.

Fixture model—(X,Y,Z value and surface normal) an uncut surface on a fixture model.

Plane—(X,Y,Z value and surface normal) a flat machined surface. Cylinders or circles cut using linear motions can also return this identity.

Ruled surface—(X,Y,Z value and surface normal) a surface cut by a multi-axis motion.

Sphere—(X,Y,Z center and radius) a surface cut by the end of a ball endmill.

Stock model—(X,Y,Z value and surface normal) an uncut surface on a stock model.

Torus—(X,Y,Z center, axis vector, major and minor radius) a fillet in a corner cut by a bull endmill, or fillet left on a diameter by a radiused cutter during a turning operation.

Torus sweep—(X,Y,Z center, axis vector and major radius) This feature is typical of an angled (ramping or plunging) surface cut by a bull endmill.

Calculating workpiece and uncut stock volumes

To calculate workpiece and uncut stock volumes:

1. Click  on the Toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Volume** tab.
3. Click on the stock model to receive volume information about.

The volume information is displayed in the X-Caliper window.

Tip:

You can use the volume value and material density to calculate the weight of the machined part.

Measuring distance or angle between two model features

VERICUT's **X-Caliper** function makes it easy to measure distance or angle between two selected model features. VERICUT automatically determines if the distance or angle value and displays it in the X-Caliper window.

To measure distance or angle between 2 model features:

1. Click  on the toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Distance/Angle** tab.
3. Choose the "From" feature:
 - In the "From" area, click the choice arrow next to the option list.
 - Choose the feature type to measure from.
 - Click the model feature to measure from-the "From" data fields are filled in.
4. Choose the "To" feature: follow the same procedure as described above for the "From" feature.

The distance or angle measurement is displayed in the X-Caliper window.

Tip:

You can edit data in the From/To fields, then press **Compute** to obtain the desired measurement.

Measuring scallop height between two cylinders

When an area has been machined using back-and-forth cutting motions, use the procedure below to measure the height of the scallop left between parallel cuts.

To measure scallop height:

1. Click  on the toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Scallop** tab.
3. Choose the "Cylinder 1" feature:
 - In the "Cylinder 1" area, click the choice arrow.
 - Choose the cylinder on one side of the scallop to measure.
4. Choose the "Cylinder 2" feature: follow the same procedure as described above for "Cylinder 1".

The scallop measurement is displayed in the **Scallop Height** field in the X-Caliper window.

Measuring from a point to the closest point on the model

To measure from a specified point to the closest point on the model:

1. Click  on the Toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Closest Point** tab.
3. Enter XYZ values of the point to measure from. (These values can not be filled via mouse pick.)
4. Click **Compute**.

The measurement is displayed in the X-Caliper window.

Measuring thickness of solid material

Use the procedure below to measure thickness of solid objects on the model, such as: walls, floors, bosses, etc.

To measure thickness of a solid model feature:

1. Click  on the toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Stock Thickness** tab.
3. Click on the solid model feature to measure for thickness.

The thickness measurement is displayed in the X-Caliper window.

Tip:

You can edit data in the XYZ or IJK fields, then press **Compute** to obtain the desired measurement.

Measuring air distance

Use the procedure below to measure air distance, such as: between pocket walls, inside hole diameters, etc.

To measure air distance:

1. Click  on the toolbar, or click **Analysis menu > X-Caliper**.
2. Click the **Air distance** tab.
3. Click on the solid model feature to measure air distance from.

The thickness measurement is displayed in the X-Caliper window.

Tip:

You can edit data in the XYZ or IJK fields, then press **Compute** to obtain the desired measurement.

Capturing VERICUT images

Image capture control in VERICUT

VERICUT provides many methods of capturing images during a simulation. Images of the workpiece being machine as well as machine tool motions can be captured and sent directly to the printer, output to a file in various formats, or recorded for later viewing in a "slideshow" or "movie". Images can be captured interactively, automatically via AutoSave, or via batch processing.

The sections that follow describe various ways you can capture VERICUT images.

What do you want to do?

- Print view images

- Output JPEG, PS, EPSF, or TIFF image files

- Record snapshots or animated images

Printing view images

Use VERICUT's **Print** function and the procedure below to print view images.

To print view images:

1. Display the image(s) in VERICUT you want to capture then click  on the Toolbar, or click **File menu > Print**.
2. In the window that opens, set the printing options you want, then click **Print**.

All views are printed. Each view is automatically fit to the printable area and printed on a separate page.

Tips:

You can save images of the VERICUT model automatically and in batch mode using the **File menu > AutoSave** and **ShadeCopy** functions.

ShadeCopy can output VERICUT images to a file for printing on other devices, or import into desktop publishing applications.

Outputting JPEG, PS, EPSF, or TIFF image files

Use the procedure below to output VERICUT images in a variety of image file formats. All view ports are captured and output to the file.

To configure VERICUT to capture images in JPEG, PostScript, EPSF, or TIFF format:

1. Click **File menu > Images > ShadeCopy**.
2. On the **Attributes** tab, choose the **Type** of image file to create, and **Method** of coloring. (Hint: **Palette Color** images look the same as **RGB Color** images, but are about 1/3rd the size.)
3. Choose the desired **Orientation**.
4. Select or type the name of the **Output File** to receive the captured image.
5. If applicable, access **PostScript Settings** to place the image on the printed page and text annotation.
6. Press **OK**.

VERICUT is now configured to capture images in the desired format. Output JPEG, PS, EPSF, or TIFF image files as described below.

To interactively output a VERICUT images:

1. Display the image(s) in VERICUT you want to capture, then click **File menu > Images > ShadeCopy**.
2. Assuming the ShadeCopy window has been configured (see above), press **Output**.

The ShadeCopy image file is saved.

To automatically output VERICUT images during processing:

1. Assuming the ShadeCopy window has been configured (see above), click **File menu > AutoSave**.
2. Click the **ShadeCopy** tab.
3. *To send images to the printer-* select **Print Command** and verify the command in the field is correct. Edit if necessary.
4. *To send images to files-* clear the **Print Command** checkbox. In the **ShadeCopy File** field, enter the base file name for saved files.
5. Click **Properties** and verify ShadeCopy settings are correct. If necessary, change them.
6. Under **Auto Save** and **Auto Error**, select the event(s) to output a ShadeCopy image file. If both are in use, be sure they have different **ShadeCopy File** names to avoid overwriting.
7. Press **OK** and continue cutting.

ShadeCopy image files are saved at the specified events.

Tips:

Help on VERICUT

You can save VERICUT images in batch mode using the **File menu > AutoSave** and **ShadeCopy** configurations described above.

File menu > Print can output VERICUT images directly to the printer.

Recording snapshots or animated images

Use the procedure below to record VERICUT images and machining status in a VERICUT Image file. All view ports are recorded.

To record VERICUT snapshots or animation:

1. Click **File menu > Images > Record**.
2. In the window that opens, select or type the **Filename** of the file to receive recorded images.
3. To append images to an existing file, select **Append**. You can add delay before appended images by entering the number of seconds of delay into the **Append Delay** field.
4. *To save a single snapshot-* press **Snapshot**.
5. *To record snapshot images automatically-* select the desired **Snapshot Events** at which to record an image (you can select more than one), then press **Record**.
6. *To record animated images-* select the desired **Animation Control** option, then press **Record**.
7. If applicable, use **Insert Stop** to insert a stop point, or **Insert Delay** to insert seconds of delay.
8. When you are finished recording, press **Stop**. Recording must be stopped before the Image file can be viewed.

All movements in the graphics area are recorded, including tool and machine motions, model manipulation such as rotating, zooming, etc. At a later time, you can play back the recorded Image file as described below.

To play back a recorded Image file:

Note: Recording must be stopped before the Image file can be viewed.

1. Click **File menu > Images > Playback**.
2. In the window that opens, select or type the name of the **Image file** to play.
3. Press **Load**.

The image player window opens with the first image displayed.

4. Set the control for playback speed as required: animation—use **Speed** slider, snapshot—enter **Snapshot Delay**. Automatic stopping is available via **Control** options.

5. Press **Play forward**  to start playing images in the forward direction.
6. When done, press **Done** (not the window "X" control) to close the image player window.

Hint: Toolbar short cut for recording and playing back VERICUT images:  

Machine and control building

Building NC machines

About building NC machines

This topic is still being written.

Some of the details will include:

- definition of "Machine coordinate system"

- definition of "Machine zero"

- overview of building NC machines

Building machines involves defining components and adding models to represent machine size and shape, as described in "About modeling in VERICUT" earlier in this chapter. To save space, the sections that follow assume you have read this section and have an understanding about modeling in VERICUT.

- **A [B,C] Turret**—Indexing tool turret that rotates about X, Y, or Z axes. Like rotary components, the turret component origin is the pivot point for rotary indexing. The presence of these components cause all connected tools to be displayed making it possible to detect collisions resulting from turret indexing.

A Turret component is commanded to move using a tool change command (e.g. Tn). The turret indexes based upon activating a Tool component (see below) connected to the turret. Tool component orientation determines the tool index position in the turret. Rotary... modals control how turrets rotate, e.g.: rotation direction, intermediate points to improve visualization of indexing, etc.

What do you want to do?

- Add, modify, or delete machine tables

- Detect collisions with machine components

- Detect axis overtravel errors

- Change how the machine moves in rapid

- Manually move the machine via MDI

About configuring tables

Tables are used to store values for control features, axis values for important machine locations, and other data related to processing G-code tool path files. VERICUT windows designed to support defining and maintaining tables, known as "Tables windows", all have similar features, and are used in a similar manner.

The following VERICUT windows provide access to tables:

Setup menu > Machine > Settings: Tables tab—Specify important machine locations and detail tool change motion.

Setup menu > G-Code > Settings: Job Tables tab—Specify work offsets and machine locations specific to the current tool path file(s).

Setup menu > G-Code > Settings: Tool Tables tab—Store tool-related offset and compensation data.

The sections that follow discuss how to use Tables window features to add new, modify existing, and delete unwanted data from tables.

Tables contain data that is stored and retrieved using a corresponding register or "index" value. Each table index may be responsible for storing one or more data values, depending on the table. An index with associated data is known as a table "row".

For table-specific information, such as data entry formats and their affect in VERICUT, refer to "Tables - listed alphabetically" in the *Reference* chapter.

Tables window features:

Tables list—List of tables that can be referenced by a specific VERICUT function. The listed tables are different, depending on the VERICUT function used to access the Tables window (see above).

Table Name—Identifies the table active for adding, or editing if it already exists. Depending on the active choice, data appears differently in the **Tables list**, and is entered differently in the fields that follow.

SubSystem ID—ID of the subsystem for which the table is being defined.

Index or Register—The index number that will be used by VERICUT to access corresponding table data. For tool tables, this number typically corresponds to a tool or offset register number. For job and machine tables, the index number may correspond to an offset register number, or an integer value, as determined by the table.

Values—One to nine numeric values, as determined by the table. Multiple-value entries require spaces between values.

In the sample window below, the table active for editing is the **Work Offset** table. The active table row is index "56", and contains the values "20 0 2.4". The table is applicable to subsystem "1" of the machine.

G-Code Settings

Variables | Subroutines | Block Skip | Turning | WireEDM

Settings | Job Tables | Tool Tables

Tables	SubSystem	Index	Values
Work Offsets	1	54	0 0 2.4
	1	55	10 0 2.4
	1	56	20 0 2.4

Table Name:

SubSystem ID:

Register:

Values (XYZABCUVW):

Add—Adds data configured in the Tables window to the active table, as specified via **Table Name**. If the table was not previously defined, then the table is also added.

Modify—Modifies the selected row in the table to have the data configured in the Tables window.

Delete—Deletes the selected row from the table. If its the last table row, then the table is also deleted.

Detecting collisions with machine components

When a 3-D NC machine is in use, you can detect collisions between machine components and the workpiece, or other components on the machine.

To configure VERICUT to detect collisions with machine components:

1. Click **Setup menu > Machine > Settings**.
2. Click the **Collision Detect** tab.
3. Configure as many **Component 1-2** collision cases as desired.
4. Use **Default Near Miss** and **Set All** to set all collision case tolerances, then edit individual collision case **Tolerance** as required.
5. Click **Add** to add the collision case to the list.
6. To start checking for collisions, select **Collision Detection On**.
7. Click **OK** or **Apply**.

Colliding components are highlighted using the red **Error** color, and errors are issued to the Log file identifying collision causing block(s) and machine components.

Detecting axis overtravel errors

After defining an NC machine, you can set the range of travel for any or all motion axes using the features on the **Setup menu > Machine > Settings: Travel Limits** tab. Once set, overtravel detection is turned on/off on the same window via the "**Overtravel Detection On**" feature.

To configure VERICUT to detect axis overtravel errors:

1. Click **Setup menu > Machine > Settings**.
2. Click the **Travel Limits** tab.
3. Modify travel limits as required for each machine motion axis, or select **Ignore** to ignore limit checking on that axis.
4. If desired, set the **Overtravel Color** to make it easier to see violating machine components.
5. To start checking for overtravel errors, select **Overtravel Detection On**.
6. Click **OK** or **Apply**.

Overtravel errors cause the violating machine component to "light up" in the **Overtravel Color** and errors are output to the Log file identifying the problem component.

Changing how the machine moves in rapid

VERICUT provides full control over how machine axes move in rapid positioning mode (e.g. G0). By default, machine axes are interpolated with equal priority to arrive at their destinations at the same time. Use the procedure below to change the priority to simulate non-interpolated rapid motion such as "dog leg" or "squared off" movement.

To change how machine axes move in rapid mode:

1. Click **Setup menu > Machine > Settings**.
2. Click the **Axis Priority** tab.
3. Configure the window as required. (See window features)
4. Click **OK** or **Apply**.

Examples follow.

Example 1- Configure "XY dog leg" rapid motion where X and Y move at a 45 degree angle until one axis reaches its destination, then the remaining axis finishes its move.

1. Select the "**X**" register in the axis priority list and clear the **Interpolated** checkbox.
2. Repeat the above step for the "**Y**" register.
3. Click **OK** or **Apply**.

Example 2- Configure "squared off Z" rapid motion: rapid XYZ motion with positive Z moves Z first, then X and Y move interpolated together. Rapid XYZ with negative Z moves X and Y (interpolated), then Z.

1. Select **Pos/Neg**.
2. Ensure **Positive** is selected to access the priorities for axes moving in the positive Z direction.
3. Select the "**X**" register in the axis priority list and choose **Priority=2**.
4. Repeat the above step for the "**Y**" register.
5. Select **Negative** to access the priorities for axes moving in the negative Z direction.
6. Select the "**Z**" register, choose **Priority=2**.
7. Click **OK** or **Apply**.

Manually moving the machine via MDI

After you define an NC machine and control, you can manually move the machine using commands typed in an "MDI console" window-similar to how an operator tests machine motions on the shop floor. This procedure can be especially helpful to test or prove-out newly constructed machines.

Note: If you made any changes to the machine/control configuration, be sure to press **Reset**  before manually moving the machine.

To manually enter and process blocks of machine code data:

1. Click  on the Toolbar, or click **Setup menu > Control > MDI**.
2. Enter the block of data to process.
3. Press **Apply** to process the data block.

You can process as many blocks of data as you wish in this manner.

Tip: To process several blocks of data at once, consider putting all the blocks into a "test program" file that VERICUT can be configured to process as a tool path file.

Building NC controls

About building NC controls

In general, **Setup > Control** menu functions are used to define how the NC control on a machine processes machine code data, also known as "G-code" data. As delivered from CGTech, VERICUT supports processing M and G-code data in EIA Std RS-274 format, as well as some conversational formats. However, using VERICUT's powerful user interface or the Machine Developer's Kit, you can add support for non-standard codes and formats. All control-related information is savable in a Control file where it can be coupled with other machines having the same or similar NC control characteristics.

The following sections provide a general overview of the control building process, and describe key terms you will find throughout related documentation.

Defining G-code words and special characters

The G-code words and special characters interpreted by the NC control are defined in VERICUT using the **Setup menu > Control > Word Format** function. Use features on this window to describe the general functionality of the character or word. In the case of words that are expected to be paired with addresses later, such as: G0 - "G" with "0", M3 - "M" with "3", "X" with any number, etc., features on this window set the type and format of the expected values.

Grouping words with addresses to perform actions

Words and a range of addresses are paired, or "grouped" using the **Setup menu > Control > Word/Address** function. Each group can then be configured to perform one or more actions via configuring them to call CGTech supplied or custom macros. For example to simulate "M0" stopping the NC machine, you would define the group as having the "M" word with "0" for the range of addresses that perform this action, then configure the group to call the "StopProgram" macro.

Group logic can also check for conditions that alter how the word/address is interpreted, such as other codes in the block, current variable values, machine states, etc. For example, "X" with a value is typically assumed to be X-axis motion. However, with "G4" also in the block, the control may need to interpret the "X" word/vale as "dwell time".

Groups that perform similar actions are arranged in a class, formerly known as a "supergroup". Classes assist VERICUT with determining the timing of when actions are performed, as described by sections that follow.

As delivered from CGTech, macros are provided to simulate most common machine and control actions. (Ref. "Macros - listed alphabetically" in the *Reference* chapter.) Macro actions can be altered, or new macros created using the Machine Developer's Kit to generate a custom CME file for the control.

Controlling timing of simulated actions

Timing of simulated actions is controlled by two things:

1. Listed order of word/address classes controls the timing of when corresponding actions are performed.

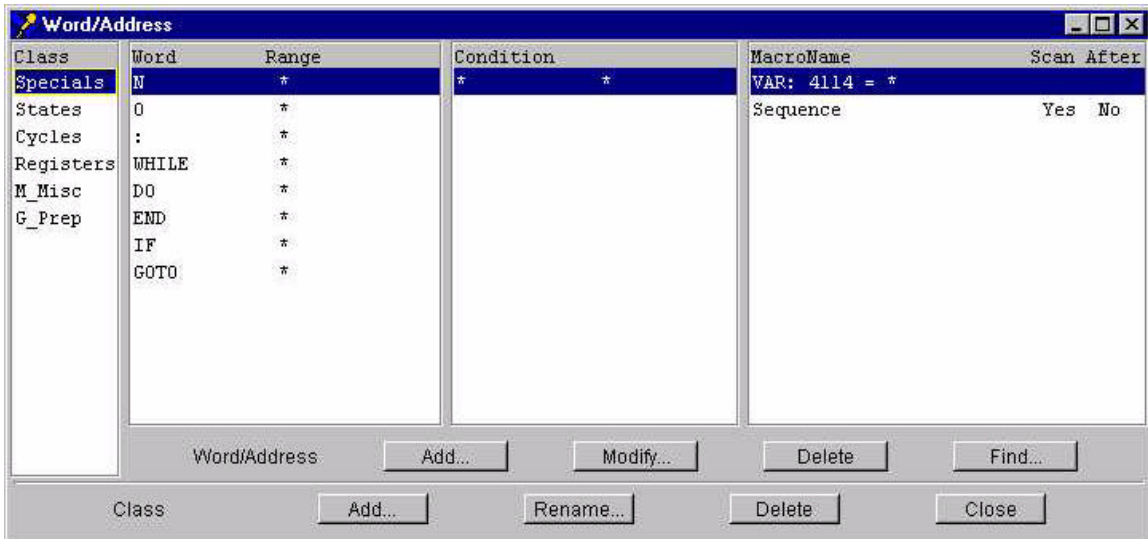
Help on VERICUT

2. For all actions associated with groups in a given class, the order in which they appear in the G-code data block controls the timing of when the actions are simulated.

An example follows:

Example of block processing by VERICUT

Consider the sample Word/Address window below and G-code data block: "N20G1X1.0Y2.0F100.0".



Following is an overview of what VERICUT does with the information in this block:

1. The block is parsed based upon control word definitions. Assuming G, X, Y, and F words are defined, our block is broken up into the following pieces, or "tokens": N20, G1, X1.0, Y2.0, F100.0
2. VERICUT checks to see if any of the block tokens can be interpreted by groups in the first class, "Specials". The "N20" token is interpreted via the "N *" group (highlighted in the figure above). Since "N" with any value is configured to call the "Sequence" macro, this code is understood as a block sequence code.

Tokens remaining to be processed: G1, X1.0, Y2.0, and F100.0

3. The next class checked is "States" which contains information about machine states, such as: motion modes (rapid, linear, circular, NURBS), active plane for machining, unit measurement system, dimension mode, etc. The "G1" token is interpreted, and calls the "MotionLinear" macro to set the linear motion state.

Tokens remaining to be processed: X1.0, Y2.0, and F100.0

4. Next, "Cycles" checked to see if there are any codes related to canned cycle processing. Since none of the remaining tokens deal with cycles, nothing occurs with this class.

Tokens remaining to be processed: X1.0, Y2.0, and F100.0

5. In the "Registers" class, all the remaining codes are interpreted. Since there is more than one group accessed in this class, macros are called and actions occur according to listed order of the tokens in the block:

Help on VERICUT

- "X1.0" calls the "XAxisMotion" macro to set "1.0" for the X-axis.
- Similarly, "Y2.0" calls "YAxisMotion" macro to set "2.0" for Y-axis location.
- "F100.0" calls the "FeedRate" macro to set "100.0" as the motion feed rate.

Tokens remaining to be processed: *NONE!*

6. With all block tokens processed, VERICUT simulates the actions related to the macros called- the machine X and Y axes move from their current location to X=1, Y=2 with a feed rate of 100.

While the above presents a fairly good overview of what occurs during block processing, it is important to note that the above scenario is simplified. There are many more things to consider in a detailed view, such as: control settings that alter how actions are performed, processing events that can trigger actions (similar to the groups discussed above), control subroutines, possible use of text string substitutions, and more.

Control settings

Functions in the **Setup menu > Control > Settings** menu provide an easy way to set the typical operation mode of the NC control. Control settings are primarily used to establish default conditions for the control. However, these functions also extend your ability to detail how VERICUT interprets specific types of machine codes, such as: circles, cycles, rotary motions, and more.

Advanced control options

The **Setup menu > Control > Adv. Options** window provides even more NC control capabilities, such as: specify subroutines available in the NC control, perform actions at key processing events (e.g. start of tool path processing, start processing a block, etc.), substitute text strings, and more. Most users do not need these advance features, but when you need them, they are there waiting for you.

What do you want to do?

Add, modify, or delete words the control interprets

Add, modify, or delete word/address classes

Add, modify, or delete word/address groups

Add, modify, or delete event-driven macros or variables

Add, modify, or delete text substitutions

Configure NC control settings- see functions in the **Setup menu > Control > Settings** menu

Learn about simulating subroutines

Test the control via manual data input (MDI)

Adding, modifying, or deleting words the control interprets

Word definitions provide the basis for VERICUT to interpret G-code characters and word addresses. When defined, word/address pairs can then be "grouped" via the **Setup menu > Control > Word/Address** function to perform specific actions by calling CGTech supplied or custom macros. Undefined words cause errors when processed, and are not acted on by VERICUT. Word definitions are savable in a Control file.

To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.



Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

Hint: Start your control configuration by choosing a control from CGTech library that is as close as possible to the control you are configuring. (See "Control files" in the *VERICUT Files* chapter.) That way, most words will be already defined for you. Often, all that is left to do is to modify a few words to match your specific control functions.

To add a word or special character to the control configuration:

1. Click **Setup menu > Control > Word Format**.
2. Click **Add**.
3. Enter the word **Name** as it appears in the tool path file.
4. Choose the word **Type** that describes the word's general function.
5. Configure the rest of the window as required to support interpreting the new word. (See window features)
6. Press **OK** or **Apply**.

To modify how a word or special character is interpreted:

1. Click **Setup menu > Control > Word Format**.
2. In the words list, highlight the word to modify.
3. Click **Modify**.
4. Modify window values as required. (See window features)
5. Press **OK** or **Apply**.

To delete a word or special character from the control configuration:

1. Click **Setup menu > Control > Word Format**.
2. In the words list, highlight the word to delete.
3. Click **Delete**.

Adding, modifying, or deleting word/address classes

Groups with similar functions are arranged into a "class". The listed order of classes in the Word/Address window helps determine the timing of when corresponding actions occur. Class, group, condition, and macro order is changed via dragging them with the right mouse button.

To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.

Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

To add a new class to the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. In the Class list, select the class to add after.
3. Press **Class: Add**.
4. Enter a unique class **Name**.
5. Press **OK** or **Apply**.

To rename an existing class in the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. In the Class list, select the class to rename.
3. Press **Class: Rename**.
4. Edit the class **Name**.
5. Press **OK** or **Apply**.

To delete a class and its groups from the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. From the Class list, select the class to delete.
3. Press **Class: Delete**.
4. Click **Yes** to confirm deleting.

Adding, modifying, or deleting word-address groups

Groups can call one or more action macros to perform various actions, or set variables. This functionality allows you to virtually "wire" a custom NC control based on machine code data to be processed. Class, group, condition, and macro order is changed via dragging them with the right mouse button.

To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.



Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

To add a new group to the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. Select the class and group to add the new group after.
3. Press **Word/Address: Add**.
4. Specify how to activate the group:
 - *To act based on a word/value:* select **Word**, then enter the **Word** name and **Range** of values. (e.g. perform an action when M10 is encountered)
 - *To act based on a variable value:* select **Variable**, then enter the **Variable** name and **Range** of values. (e.g. perform an action when #100=1)
5. Specify if a condition is to be checked for:

The default **Condition: Word, Conditional Word=*, Conditional Range=*** ignored condition checking.

 - *To check for another word/value in the block:* select **Condition: Word**, then enter the **Conditional Word** name and **Conditional Range** of values.
 - *To check a machine state:* select **Condition: State**, then pick the **Conditional State/Value** from the pick list.
 - *To check another variable value:* select **Condition: Variable**, then enter the **Conditional Variable** name and **Conditional Range** of values.
6. Configure the group action:

To call a macro: select **Macroname**, then enter or select the macro to call.

To set a variable: select **Variable**, then enter the variable **Override Value**.

The remaining window features are seldom used. For descriptions, see "Add/Modify Word/Address window features" under the **Setup menu > Control > Word/Address** section.
7. Press **OK** or **Apply**.

For descriptions of macros and actions they perform, see "Macros - listed alphabetically" in the *Reference* chapter.

To modify an existing group in the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. Select the class and its group to modify.
3. Press **Word/Address: Modify**.
4. Modify window values as required. (See window features)
5. Press **OK** or **Apply**.

To delete a group from the control configuration:

1. Click **Setup menu > Control > Word/Address**.
2. Select the class and its group to delete.
3. Press **Word/Address: Delete**.
4. Click **Yes** to confirm deleting.

Adding, modifying, or deleting event-driven macros or variables

Similar to calling macros and setting variables via words/addresses, you can also perform these actions based on key events that occur during tool path file processing. Event driven macro calls/variable assignments are configured in a similar fashion as in the **Setup menu > Control > Word/Address** window—please review that section first. Macro/variable order is changed via dragging them with the right mouse button.

To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.

IMPORTANT! The macro events listed below are required for G-code processing and should NOT be disturbed.

Start of Processing => *ConversionInit*

End of Processing => *ConversionFinish*

Start of Block Processing => *BlockInit*

End of Block Processing => *BlockFinish*



Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

To add a new event-driven action to the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Events** tab.
3. Choose the **Event** to call the macro or set the variable.
4. Configure the event action:

To call a macro: select **Macroname**, then enter or select the macro to call.

To set a variable: select **Variable**, then enter the variable **Override Value**.

The remaining window features are seldom used. For descriptions, see "Add/Modify Word/Address window features" under the **Setup menu > Control > Word/Address** section.

5. Press **OK** or **Apply**.

For descriptions of macros and actions they perform, see "Macros - listed alphabetically" in the *Reference* chapter.

To modify an existing event-driven action in the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Events** tab.
3. Select the event action to modify.

Help on VERICUT

4. Press **Modify**.
5. Modify window values as required. (See window features)
6. Press **OK** or **Apply**.

To delete an event-driven action from the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Events** tab.
3. Select the event action to delete.
4. Press **Delete**.

Adding, modifying, or deleting text substitutions

Using substitutions lets you to alter how VERICUT "sees" data in a block without editing the G-code tool path file. All substitutions occur prior to parsing the words and values in the G-code data block.

To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.

To add a text substitution to the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Substitute** tab.
3. Enter the **Input Text** (to be replaced) and **Output Text** (to replace with) you want VERICUT to use in its place.
4. Press **Add**.

To modify an existing substitution in the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Substitute** tab.
3. Select the substitution to modify.
4. Modify **Input Text** / **Output Text** as required.
5. Press **Modify**.

To delete a substitution from the control configuration:

1. Click **Setup menu > Control > Adv. Options**.
2. Click the **Substitute** tab.
3. Select the substitution to delete.
4. Press **Delete**.

Finding a group in the control configuration

Use the procedure below to find a group, either by its word and address value, or by a macro it calls. To learn more about how VERICUT processes G-code data, see "About building NC controls" in this chapter.

To find a group associated with specific G-code word/address value:

1. Click **Setup menu > Control > Word/Address**.
2. Press **Word/Address: Find**.
3. Click the **Words/Range** tab.
4. From the **Words** list, select the word of the group to find.
5. Enter the **Range** value of the word address.
6. Press **First**.

To find a group associated with macro:

1. Click **Setup menu > Control > Word/Address**.
2. Press **Word/Address: Find**.
3. Click the **Macronames** tab.
4. From the **Macroname** list, select the macro associated with the group to find.
5. Press **First**.

About simulating subroutines

Subroutines, or "subs" for short, contain additional programming or machine commands, and can exist within a G-code tool path file or reside in files separate from the main G-code program. VERICUT supports processing subs in either location. This section discusses mainly the latter- subs that reside in external files. Subroutine window features are also described in a general sense- how you use them to add, rename, and delete names of external files containing unwanted subs.

The following VERICUT windows provide access to external subroutine files:

Setup menu > Control > Adv. Options: Subroutines tab—Lists "control subroutine" file names.

Setup menu > G-Code > Settings: Subroutines tab—Lists "job subroutine" file names.

Both of these subroutine types are discussed later in this document.

VERICUT can access subs and execute the commands they contain when configured to do so. For subs to be processed correctly, VERICUT must be set to scan tool paths for branch locations. Then, to process sub calls, the VERICUT Control file must have the appropriate G-code data word/addresses configured to call the "CallNCMacro" or "CallSub" macro. Additional word/addresses need to be configured for "return to main program" commands, and variable passing if your sub calls contain variables. Ref. "Macros - listed alphabetically" in the *Reference* chapter for more information.

When a sub call is encountered, VERICUT searches the following locations (in listed order) to find a matching branch location:

1. Search the remainder of the current tool path file.
2. If not found, search external subroutines in files listed on the **Setup menu > G-Code > Settings: Subroutines** tab.
3. If still not found, search external subroutines in files listed on the **Setup menu > Control > Adv. Options: Subroutines** tab.

VERICUT will branch to the first location found to match that specified in the call. If a matching branch location is not found in the main program or any of the subs, an error message is issued.

One sub can call another, known as "nesting" subroutines. Up to 25 levels of calling subs are supported.

Hint: Debug messages can be output about calls made to subs during processing via the **Debug Loop Logic** feature on the **Setup menu > G-Code > Process Options > Debug** window.

Types of subroutines

VERICUT recognizes two types of external subs, as described below. Subs are handled differently, depending on the sub type.

1. **Control subroutines**—Subs that reside in the NC control. These subs are typically not accessible to programmers and do not change from job to job.

Processing within a control sub is not user controlled. You can not stop processing in, or step through a control sub. Only the control sub's call record can be seen in the **Info menu > Toolpath** window, and may appear in VERICUT's Log files.

Help on VERICUT

2. **Job subroutines**—Subs written by, or available to programmers that can be changed on a per job basis.

Processing within a job sub is user controlled, just like the main program tool path file. The sub call and all other job sub records can be seen in the **Info menu > Toolpath** window, and may appear in VERICUT's Log files.

Subroutine window features

File Names list—List of external files containing subs accessible to VERICUT.

Add—Opens a window to add the name of an external file to the subroutine file names list.

Replace—Opens a window to replace a file in the subroutine file names list.

Delete—Opens a window to delete names of files from the subroutine files list.

Clear—Clears (deletes) all file names from the subroutine files list.

Configuring "Circles" NC control settings

Use the procedure below to configure settings for interpreting circle center (e.g. IJK) and helical motion data.

To configure circle/helical motion settings in the control:

1. Click **Setup menu > Control > Settings > Circles**.
2. Configure the window as required. (See window features)
3. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Cutter Compensation" NC control settings

Use the procedure below to configure how the programmed tool path is compensated when cutter diameter compensation, or "CDC" is used (e.g. G41-42). When CDC is active, the programmed tool path is simulated just as it will be compensated by the NC control.

To configure the control to simulate cutter compensation:

1. Click **Setup menu > Control > Settings > Cutter Compensation**.
2. Configure the window as required. (See window features)
3. Press **OK**.
4. Enter corresponding offset values into a Cutter Compensation table.
5. To activate CDC for simulation: ensure "**Process Cutter Comp.**" is selected on the **Setup menu > G-Code > Settings: Settings** tab.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Cycle" NC control settings

Use the procedure below to configure settings for interpreting tool axis, or "canned" cycle motion blocks (e.g. G8n...).

To configure canned cycle settings in the control:

1. Click **Setup menu > Control > Settings > Cycles**.
2. Configure the window as required. (See window features)
3. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "General" NC control settings

Use the procedure below to configure general guidelines for how the control processes machine codes and number equations.

To configure general settings in the control:

1. Click **Setup menu > Control > Settings > General**.
2. Choose the **Default Machine Mode** (especially important for multi-operational machine, such as a mill-turn machining center).
3. Set the **Control Type**-choose **Generic** for Fanuc and most other controls. Other options are: **NUM** (French-made control), or **Heidenhain Conversational**.
4. Set the **Calculation Tolerance** for evaluating mathematical evaluations.
5. Choose the **Order of Math Operations** (normally **Rules of Precedence**).
6. If NC data blocks begin a number instead of a letter (example "50G01X5"), specify a **Default Word** so VERICUT interprets the number as a sequence numbers, for example "N".
7. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Motion" NC control settings

Use the procedure below to configure settings in affect when the control and machine are initially powered on, such as: default NC control motion states, and the precision used when outputting calculated values.

To configure default motion and output precision settings in the control:

1. Click **Setup menu > Control > Settings > Motion**.
2. Configure "Default" options applicable to the control: default motion type, plane selection, etc.
3. To start the machine at a location other than machine zero, ensure **Output Initial Spindle Location** is active. Enter the initial spindle location values into an Initial Machine Location table.
4. If applicable, set appropriate motion types to cancel canned cycles: rapid, linear, CW, etc.
5. Configure **Output Rapid Motions as** and **Maximum Inverse Time Output Feedrate** as required to process rapid motions.
6. Set the **Output Precision** for interpreting calculated values, such as: CDC offset positions, mathematical operations, etc.
7. If required, enter a value for **Specified Output Precision** that matches what then NC control uses for these calculations.
8. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Offsets" NC control settings

Use the procedure below to configure an initial work offset, or “fixture offset”, to be in effect at the start of tool path processing.

To configure an initial work offset:

1. Click **Setup menu > Control > Settings > Offsets**.
2. Ensure **Initial Work Offset** is active, then enter the **Initial Work Index** (e.g. 54-59) containing the initial offset to apply.
3. Press **OK**.
4. Enter corresponding offset values into a Work Offsets table.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "OptiPath" NC control settings

By default, G-code tool path blocks optimized by OptiPath receive the same sequence number, if any, that appears on the original block processed. This typically results in multiple blocks having the same sequence number.

Use the procedure below to configure settings for controlling block sequencing and word/value spacing in the optimized output tool path file.

To re-sequence or add spaces to optimized G-code tool path records:

1. Click **Setup menu > Control > Settings > OptiPath**.
2. *To re-sequence optimize tool path records:* ensure **Auto Sequence Numbering** is active, then enter the **Sequence Increment Value** with which to start and increment block sequences.
3. *To add a space between word/address pairs:* activate **Word Value Spacing**.
4. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

For more information on functions used to optimize tool path files, see **OptiPath menu > Control** and **OptiPath menu > Manager** in the *Reference* chapter.

Configuring "Rotary" NC control settings

Use the procedure below to configure settings for rotary motion commands (e.g. ABC).

To configure rotary settings in the control:

1. Click **Setup menu > Control > Settings > Rotary**.
2. Configure the window as required. (See window features)
3. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Subroutines" NC control settings

Use the procedure below to configure how subroutine names are referenced by the control.

To configure how subroutine names are referenced by the control:

1. Click **Setup menu > Control > Settings > Subroutines**.
2. Choose the **Type of Subroutine Names** which corresponds to how the control references subroutine names: **Numeric**—use numeric value (example: “O0010” is interpreted the same as “O10”), or **Text**—use text string value (example: “O0010” is not the same as the name “O10”).
3. Press **OK**.

Tips:

K&T Gemini controls- try the **Text** type of subroutine names for this control.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Tooling" NC control settings

Use the procedure below to configure settings for default tooling conditions and tool change activity.

To configure tooling and tool change settings in the control:

1. Click **Setup menu > Control > Settings > Tooling**.
2. Configure the window as required. (See window features)
3. For preselect tooling, choose **Tool Number Method=Select Only**.
4. Press **OK**.
5. If **Tool Change Retract Method** was set to retract any machine axes, enter the corresponding location values into a Tool Change Location table.
6. If **Tool Change Retract Method=Use Retraction Table**, configure the Tool Change Retraction table to with "1s" for axes to retract, and "0s" for axes that do not move during a tool change.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

Configuring "Wire EDM" NC control settings

Use the procedure below to configure settings for wire EDM machining.

To configure wire EDM settings in the control:

1. Click **Setup menu > Control > Settings > Wire EDM**.
2. Configure the window as required. (See window features)
3. Press **OK**.

Update the Control file to save the settings as part of the NC control configuration. (Ref. **Setup menu > Control > Save**)

[Help on VERICUT](#)

Optimizing tool paths

About optimizing tool paths

This topic is still being written.

Chapter 3 Reference

Map of 4.4 functions in 5.0

The VERICUT 4.4 functions and features that moved or are not available in 5.0, are listed below by 4.4 menu location. For ease of reference, 4.4 VERICUT and Machine Simulation features separated into different lists.

VERICUT 4.4 features => VERICUT 5.0

Machine Simulation 4.4 features => VERICUT 5.0

VERICUT 4.4 features => VERICUT 5.0

VERICUT 4.4 user interface location:	VERICUT 5.0:
File>Open>In Process	File>In Process>Open
File>Open>User	File>Open
File>Save>In Process	File>In Process>Save
File>Save>User	File>Save
File>Save As>In Process	File>In Process>Save As
File>Save As>User	File>Save As
File>Colors>Assignment	File>Colors: Assign tab
File>Colors>Definition	File>Colors: Define tab
File>Colors>Assignment: Fixture, Stock, Design	Model>Component Tree: select component, Edit>Component Attributes, Color
File>Properties: Database	Not available
File>Properties: Window Size	Select & drag VERICUT window border
File>Properties: Display Resolution	File>Properties>General: Display Resolution
File>Properties: Shade Smoothness	No longer needed
File>Properties: Model Boundary	Automatically calculated
File>Report>Summary	Info>File Summary
File>Report>Detailed	Info>Control Report/G-Code Report

Help on VERICUT

File>Export>FastMill Model	File>Export Model (4.4 export "Surface" use 5.0 export "IGES")
File>Export>VERICUT Model	File>Export Model
File>ShadeCopy	File>Images>ShadeCopy
File>ShadeCopy: Command	File>AutoSave: ShadeCopy tab, Print Command
File>Auto Save>FastMill Model	Use: File>AutoSave: In Process File
File>Log>Name	Info>VERICUT Log: File>Save As
File>Log>Reset	Info>VERICUT Log: File>Reset
View>Orient: Dyanamic=Off	View>Orient (or Toolbar): click dynamic icon again to turn off (raise)
View>Orient: Pan	Use: dynamic Pan
View>Orient: Redraw	View>Orient: Refine
View>Attributes: Axis Limits	Not available
View>Attributes: Line Filter	Not available
View>Attributes: View Boundary	No longer applies
Model>Model Definition: STL, Normals	Use: File>Properties: General tab, Stock Consistency Check (Stock models) or File>Convert>PolyFix (all other model types)
Model>Model Definition: Cylinder/Cone-Tolerance	File>Properties: Tolerance tab, Model Tolerance
Model>Model Options	Not needed
Model>Model Orientation	Model>Component Tree; Edit>Component Attributes: Position
Toolpath>Toolpath Control	Setup>Toolpath
Toolpath>Toolpath Control: Multiple Toolpath Files	Setup>Toolpath: Add
Toolpath>Toolpath Control: Machining Type	Setup>Motion: Machining Type (Turning planned in a future release)
Toolpath>Toolpath Control: Toolpath Type=MCD RevPost I	Use: Setup>Toolpath>Toolpath Type=G-Code Data
Toolpath>Toolpath Control: Units	Setup>APT Settings: Default Toolpath Units
Toolpath>Toolpath Control: Cut Control	Setup>Motion: Motion
Toolpath>Toolpath Control: Settings (APT)	Setup>APT Settings
Toolpath>Toolpath Control: Settings (G-code)	Setup>G-code>Settings

Help on VERICUT

Toolpath>Toolpath Control: Max Distance	Setup>Motion: Animation Speed
Toolpath>Toolpath Control: Max Angle	Setup>APT Settings: Max Tool Axis Angle
Toolpath>Toolpath Control: Reset Angle	Setup>APT Settings: Tool Axis Reset Angle
Toolpath>Toolpath Control: APT Cycle Def	Setup>APT Settings: Cycle Def
Toolpath>Toolpath Orientation	Setup>Local Coord. System
Tools>Tool Control: Tool Change Control	Setup>Toolpath: Tool Change By
Tools>Tool Control: Shank	Setup>Toolpath: Use Holders
Tools>Tool Control: Holder	Setup>Toolpath: Use Holders
Tools>Tool Control: Tool Library	Setup>Tool Manager>File
Tools>Tool Control: Tool Properties	Setup>Motion: Tool
Tools>Tool Control: Display	Setup>Motion: Display
Tools>Tool Control: Cut Color Method	Setup>Motion: Cut Color Method
Tools>Tool Control: Control Point	Setup>Motion: Control Point
Tools>Tool Control: Calculate Cutter Height	Setup>Motion: Calculate Min. Cutter Extension
Tools>Tool Control: Clearance	Setup>Motion: Holder Clearance
Tools>Tool Control: Minimum Height	Setup>Motion: Min. Cutter Height
Tools>Tool Control: Use Tool List	Setup>Toolpath: Tool Change By>List
Tools>Tool Control: Cut Vector	Seen only in FastMill cut mode
Tools>Tool Control: Resolution	Automatically calculated
Tools>Tool Manager	Setup>Tool Manager
Analysis>History	Analysis>X-Caliper: Feature/History
Analysis>Volume	Analysis>X-Caliper: Volume tab
Analysis>X-Caliper: Interrogation Type=Distance/Angle, Send Screen Pick Data	Replaced by "choice arrows" on Analysis>X-Caliper: Dist/Angle tab
Analysis>X-Caliper: Interrogation Type=Distance/Angle, lock	Not available
Analysis>Toolpath Min/Max	Replaced by Model>Model Definition: Model tab, Type=Block, Autoblock
Analysis>Translucency	Planned in a future release
Analysis>Section	View>Section

Help on VERICUT

Applications>AUTO-DIFF	Analysis>AUTO-DIFF
Applications>AUTO-DIFF>Compare: Error List	Analysis>AUTO-DIFF: Report
Applications>AUTO-DIFF>Compare: Write Report	Automatically created - view via Analysis>AUTO-DIFF: Report
Applications>AUTO-DIFF>Compare: Display	Model>Model Definition, Component Attributes (having this window open shows design models)
Applications>AUTO-DIFF>Compare, Settings: In-Process File	Planned in a future release
Applications>AUTO-DIFF>Compare, Settings: Show	Analysis>AUTO-DIFF, Settings: Comparison Type
Applications>AUTO-DIFF>Constant Gouge	Planned in a future release
Applications>AUTO-DIFF>Other: Model Load Order	Not available
Applications>AUTO-DIFF>Other: Protect Cuts	Analysis>AUTO-DIFF: Cut Options tab
Applications>OptiPath>Control	OptiPath>Control
Applications>OptiPath>Control: OptiPath=On/Off	OptiPath>Control: On/Off/Interactive
Applications>OptiPath>Control: Setup	OptiPath>Control: Material/Machine option lists; tool list Xrefs in Setup>Toolpath: Use Tool List
Applications>OptiPath>Control: Multi Axis	Automatically controlled by VERICUT
Applications>OptiPath>Control: Accel/Decel	OptiPath>Manager: Accel/Decel
Applications>OptiPath>Control: Data Link	OptiPath>DataLink
Applications>OptiPath>Manager	OptiPath>Manager
Applications>OptiPath>Manager: Add, Calculator	OptiPath>Manager: Feed/Speed tab, unselect all optimization methods & adjust sliders as needed
Applications>OptiPath>Manager: Add, Test	OptiPath>Manager: Feed/Speed tab, move Cutting Condition sliders as needed
Applications>OptiPath>Manager: Add, Reference Data	Not available
Applications>Slice	Planned in a future release
Applications>Machine Simulation	View>Attributes>View Type=Machine

Machine Simulation 4.4 features => VERICUT 5.0

Machine Simulation 4.4 user interface location:	VERICUT 5.0:

Help on VERICUT

View>Attributes: Axis Display	View>Axes
View>Attributes: Axis Limits	Not available
View>Attributes: Primitive Origin	View>Axes>Model
View>Attributes: Sectioning	View>Section
View>Attributes: Line Filter	Not available
View>Attributes: View Boundary	No longer applies
Info>Messages	All messages are sent to the VERICUT main window message area
Info>Output	Setup>G-code>Process Options>View APT Output
Info>Log	Setup>G-code>Process Options>View APT Log
Job>Settings	Setup>G-Code>Settings: Settings tab
Job>Settings: Programming Method	Setup>G-Code>Settings: Settings tab
Job>Settings: Output File	Setup>G-Code>Process Options>APT Output
Job>Settings: Log File	Setup>G-Code>Process Options>APT Output
Job>Settings: Simulation	Setup>Machine>Settings: Machine Simulation On
Job>Settings: Conversion	Setup>G-Code>Process Options>APT Output
Job>Settings: Conversion Method=Scan On/Scan Off	Setup>G-Code>Settings: Settings tab, Scan Toolpath Files
Job>Settings: Conversion Method=Column Ignore (Scan off)	Setup>G-Code>Settings: Settings tab, Ignore Data after Column
Job>Settings: Default Tolerance	File>Properties: Tolerance tab, Interpolation Tolerance
Job>Subroutines	Setup>G-Code>Settings: Subroutines tab
Job>Models	Model>Model Definition: Model tab
Job>Models: Primitives, Cylinder/Cone-Tolerance	File>Properties: Tolerance tab, Model Tolerance
Job>Tables	Setup>G-Code>Settings: Job Tables tab
Job>Variables	Setup>G-Code>Settings: Variables tab
Job>Collision	Setup>Machine>Settings: Collision Detect tab
Job>MDI	Setup>Control>MDI
Tools>Tables: Tool Index Table	Setup>Toolpath: Tool Change By=List + Use Tool List

Help on VERICUT

Tools>Tables: all other tables	Setup>G-Code>Settings: Tool Tables tab
Machine>Open, Save, Save As	Setup>Machine>Open, Save, Save As
Machine>Tables	Setup>Machine>Settings: Tables
Machine>Travel Limits	Setup>Machine>Settings: Travel Limits tab
Machine>Priority	Setup>Machine>Settings: Axis Priority tab
Machine>Components	Model>Component Tree
Machine>Components: Add	Model>Component Tree: Component>Insert/Append
Machine>Components: Modify	Model>Component Tree: Edit: Component Attributes
Machine>Components: Delete	Model>Component Tree: Edit: Delete
Machine>Components: Primitives, Cylinder/Cone- Tolerance	File>Properties: Tolerance tab, Model Tolerance
Control>Open, Save, Save As	Setup>Control>Open, Save, Save As
Control>CME File	Setup>Control>Adv. Options: Dev Kit CME tab
Control>Subroutines	Setup>Control>Adv. Options: Subroutines tab
Control>Words	Setup>Control>Word Format
Control>Groups	Setup>Control>Word/Address
Control>Events	Setup>Control>Adv. Options: Events tab
Control>Substitutions	Setup>Control>Adv. Options: Substitute tab
Modals	Setup>Control>Settings
Modals>Block Skip	Setup>G-Code>Settings: Block Skip tab
Modals>Circles	Setup>Control>Settings>Circles
Modals>Cutter Compensation	Setup>Control>Settings>Cutter Compensation
Modals>Cutter Compensation: Type of Cutter Compensation	Setup>G-Code>Settings: Process Cutter Comp.
Modals>Cycles	Setup>Control>Settings>Cycles
Modals>Cycles: Cycle Motion	Setup>Motion: Drill Cycle
Modals>Debug	Setup>G-Code>Process Options>Debug
Modals>Debug>Debug Look Ahead Cutter Compensation	Setup menu>G-Code>Process Options>Debug: Debug Cutter Compensation

Help on VERICUT

Modals>General	Setup>Control>Settings: General
Modals>General: Max Errors/Max Warnings	Setup>Motion: Stop at Max Errors/Stop at Max Warnings
Modals>General: Stop on Program Stop/Stop on Optional Stop	Setup>Motion: Stop at
Modals>General: Max Distance	Setup>Motion: Animation Speed
Modals>Messages	Setup>G-Code>Process Options>Messages
Modals>Offsets	Setup>Control>Settings>Offsets
Modals>OptiPath	Setup>Control>Settings>OptiPath
Modals>Rotary	Setup>Control>Settings>Rotary
Modals>Rotary: Rotary Motion Tolerance	Setup>G-Code>Settings, Settings tab
Modals>Rotary: Rotary Rapid Motion Tolerance	Setup>G-Code>Settings, Settings tab
Modals>Subroutines	Setup>Control>Settings>Subroutines
Modals>Tooling	Setup>Control>Settings>Tooling
Modals>Tooling: Initial Tool/Initial Tool Number	Setup>Toolpath
Modals>Wire EDM	Setup>Control>Settings>Wire EDM
Modals>Wire EDM: Wire Diameter	Setup>G-Code>Settings: Wire EDM tab
Modals>Wire EDM: Work Table to XY Plane/XY plane to UV Plane	Setup>G-Code>Settings: Wire EDM tab

VERICUT Functions Reference

VERICUT functions - listed by menu

The VERICUT functions are listed below by menu, as they appear in the user interface. See the appropriate section for details.

File menu functions—These functions open and save the most common files used in VERICUT, access converter products, and control display colors and properties, and export cut model data.

View menu functions—These functions set up the number of views, orientation (angle and distance) for each view, and store or select commonly used views.

Info menu functions—These functions access session information, such as the current tool path file, machining status, Log and Results files.

Model menu functions—These functions provide import and modeling capabilities to define the workpiece, holding fixtures, and NC machines that will be used in the verification session.

Setup menu functions—These functions are used to setup the tool path file(s) to be processed, supply cutting tool descriptions, as well as define NC machines/controls for G-code tool path simulations.

Analysis menu functions—These functions are used to analyze and inspect the VERICUT model.

OptiPath menu functions—These functions access OptiPath-the feed rate and spindle speed optimization software that creates "optimized" tool paths that cut parts in the least amount of time.

Hint: To find functions in alphabetical order, use the Index.

File menu functions

New

Open

Save

Save As

In Process > Open, Save, Save As

Working Directory

Convert > IGES, PolyFix, Surface to Solid, VDAFS, Binary CL, Mastercam

Export Model

Edit > Toolpath File, Text File

Colors

Help on VERICUT

Properties

Print

Images > Record, Playback, ShadeCopy

AutoSave

Exit

View menu functions

Layout

Orient

Attributes

Section

Select/Store

Axes

Toolbar

Look & Feel

Info menu functions

Toolpath

Status

File Summary

G-Code Report

VERICUT Log

Model menu functions

Model Definition

Component Tree

Delete Material

Setup menu functions

Toolpath

Motion

Help on VERICUT

G-Code > Settings

G-Code > Process Options > Output Files, Messages, Debug, User Defined, View APT Output,
View APT Log

APT Settings

Coord. System

Tool Manager

Control > Open, Save, Save As, Word Format, Word/Address, MDI, Modals, Adv. Options

Machine > Open, Save, Save As, Settings

Analysis menu functions

X-Caliper

AUTO-DIFF

OptiPath menu functions

Control

Manager

DataLink

File menu functions

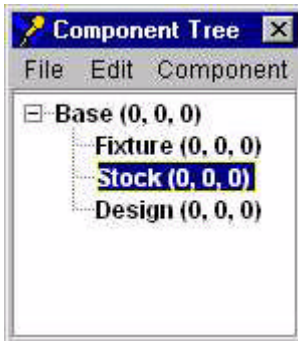
New

Location: **File menu > New**

Sets VERICUT to a "new" condition, as if no files have been loaded. Afterwards, configure VERICUT by adding models, tool paths, descriptions of cutting tools, etc.

To establish the new condition, VERICUT purges all data from memory, then opens the init.usr (inch session) or initm.usr (millimeter session) User file from the CGTech library. The components shown below are automatically defined, ready for you to add models.

Component Tree after a new condition:



Tips:

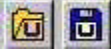
You can customize how VERICUT establishes the new condition by making changes to the init.usr/initm.usr files. Store the changed files in your working directory, or the CGTech library directory.

If needed, change the units for the new session as follows:

- Click **File menu > Properties**.
- On the **General** tab, select the desired **Default Units: Inch** or **Millimeter**.

Open (User file)

Location: **File menu > Open**

Toolbar short cut for saving and opening User files: 

Opens a window to open (load) a User file. User files contain VERICUT session settings, known as "user configuration values" or "user values", required for processing a tool path file.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names. To save space, the features described below are unique to VERICUT file selection windows or this window.

Open User File window features:

Unit Conversion—When the chosen unit value is different from the current session units, measurement values in the file being opened are converted to the specified units.

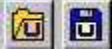
Notes:

Add-user and IP files can also be opened using this function. However, when an IP file is opened only the User file portion is loaded. To load the IP file with its model database, use the **File menu > In Process > Open** function.

Environment variables can be used to affect VERICUT's use of User files. For more information, see "Environment variables" in the *Reference* chapter.

Save (User file)

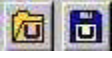
Location: **File menu** > **Save**

Toolbar short cut for saving and opening User files: 

Saves (updates) an existing User file with the current VERICUT session settings.

Save As (User file)

Location: **File menu** > **Save As**

Toolbar short cut for saving and opening User files: 

Opens a window to save a User file. User files contain VERICUT session settings, known as "user configuration values" or "user values", required for processing a tool path file. Note that the only model data saved in a User file is a description of the beginning stock model. To save 3-D model data, use **File menu** > **In Process** > **Save As**.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names. To save space, the features described below are unique to VERICUT file selection windows or this window.

Save User File window features:

Units to Save—Controls the units (Inch or Millimeter) in which to store User file data. If the session units are different than the units to save, then values are converted accordingly when stored in the new file.

Note: Environment variables can be used to affect VERICUT's use of User files. For more information, see "Environment variables" in the *Reference* chapter.

Open (In Process file)

Location: **File menu > In Process > Open**

Toolbar short cut for saving and opening IP files: 

Opens a window to open (load) an In Process file, or "IP file". IP files contain all the data necessary to re-establish a VERICUT session, including the current VERICUT model (with cuts), user interface settings, and a copy of the Log file.

Features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names.

Note: Environment variables can be used to affect VERICUT's use of IP files. For more information, see "Environment variables" in the *Reference* chapter.

Save (In Process file)

Location: **File menu** > **In Process** > **Save**

Toolbar short cut for saving and opening IP files: 

Saves (updates) an existing In Process file, or "IP file", with the current VERICUT model, including all machine cuts.

Save As (In Process file)

Location: **File menu** > **In Process** > **Save As**

Toolbar short cut for saving and opening IP files: 

Opens a window to save an In Process file, or "IP file". IP files contain all the data necessary to re-establish a VERICUT session, including the current VERICUT model (with cuts), user interface settings, and a copy of the Log file.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names.

Note: Environment variables can be used to affect VERICUT's use of IP files. For more information, see "Environment variables" in the *Reference* chapter.

Working Directory

Location: **File menu** > **Working Directory**

Opens a window to set the working directory for opening and saving VERICUT files. After setting the working directory (see how), VERICUT will look here by default whenever files are saved or opened. You can also quickly access files in this directory via the **Working Directory** entry in the file selection window **Shortcut** list.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names. In this case, all that is required is to select a directory.

IGES (Converter)

Location: **File menu > Convert > IGES**

The IGES Converter converts geometry conforming to the IGES (Initial Graphics Exchange Specification) 5.0 specification to Stereolithography or VERICUT Polygon model files. (See how)

This converter is used to import complex castings, clamps, fixtures, and other 3-D design models into VERICUT. IGES data can also be used with AUTO-DIFF to verify the machined part is as expected. IGES data can be converted interactively in VERICUT, via batch processing, or via running the IGES Converter externally ("iges2v" command file).

IGES Converter window features:

Input File—Name of the file containing the IGES data to convert.

Output File—Name of the file to receive converted geometry.

Output File Type—Type of output file to create. Options are: **VERICUT** or **STL**.

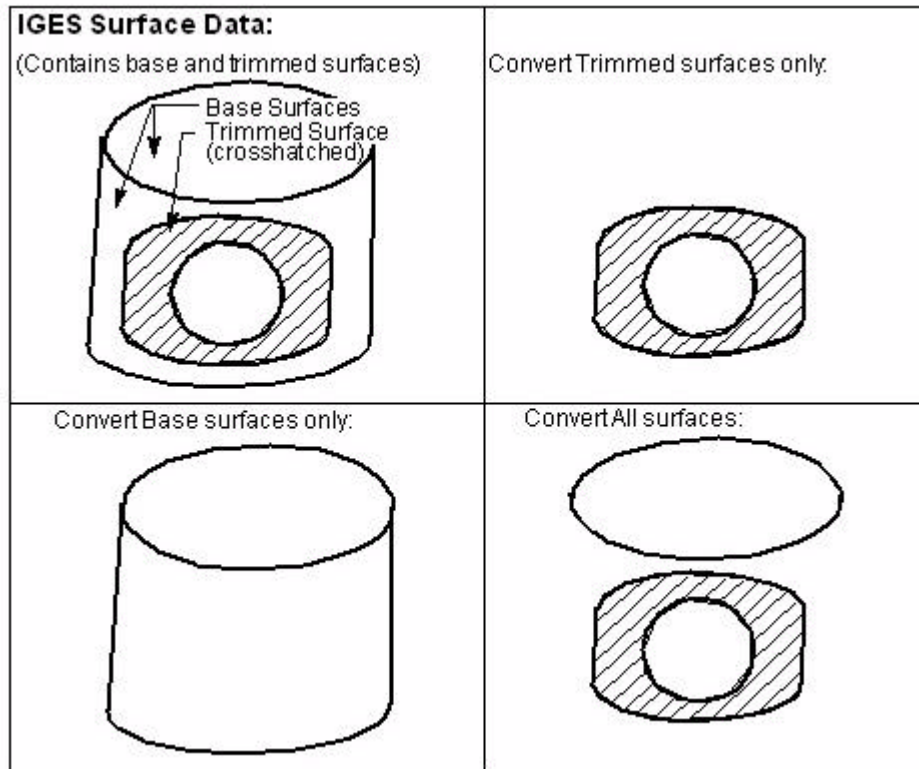
Output File Format—Format of the output file. Options are: **BINARY** or **ASCII**.

Normal Direction—The direction in which surface normal vectors point, as viewed in the CAD system. Feature only applicable to VERICUT output file types. Options are: **OUTWARD**, **INWARD**, or **INCONSISTENT**.

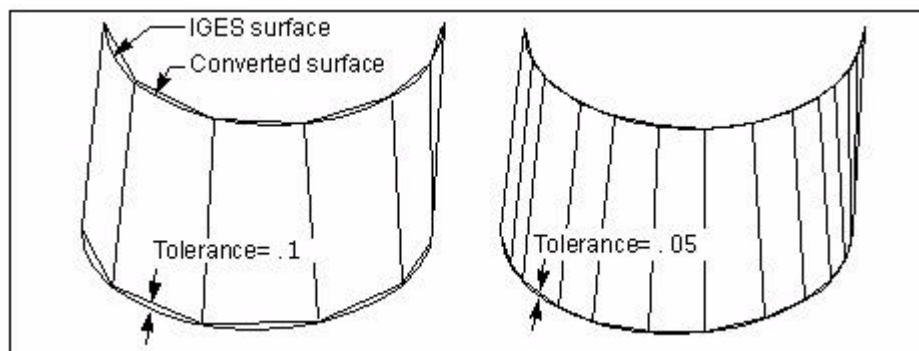
Hint: When expecting to use IGES data for solid models in VERICUT, ensure that surface normals point unidirectionally outwards in the CAD model prior to outputting the IGES data.

Surface Input Mode—Controls which IGES surfaces to convert. Options are:

- **ALL**—Convert all surfaces. With this mode, base surfaces associated with trimmed surfaces are only converted as trimmed surfaces; not in addition to trimmed surfaces.
- **BASE**—Convert base surfaces only.
- **TRIMMED**—Convert trimmed surfaces only.



Tolerance—Specifies the amount of chordal deviation allowed in 3-D space from the IGES surface when creating the converted surface. The converted surface is approximated using "facets", or triangles.



Process—Processes (converts) the IGES data according to the current window settings. Converted geometry is written to the **Output file**. Error and informational messages from the process are sent to the IGES Converter window message area, and to a log file named "iges2v.log" created in the working directory.

Summary—Prints a summary of the **Input file** contents to the IGES Converter window message area.

Clear—Clears the IGES Converter window message area.

Tips:

The **Stock Consistency Check** option is recommended for use with IGES Stock models. (Ref.

Help on VERICUT

File menu > Properties: General tab)

What do you want to do?

See how to convert IGES data

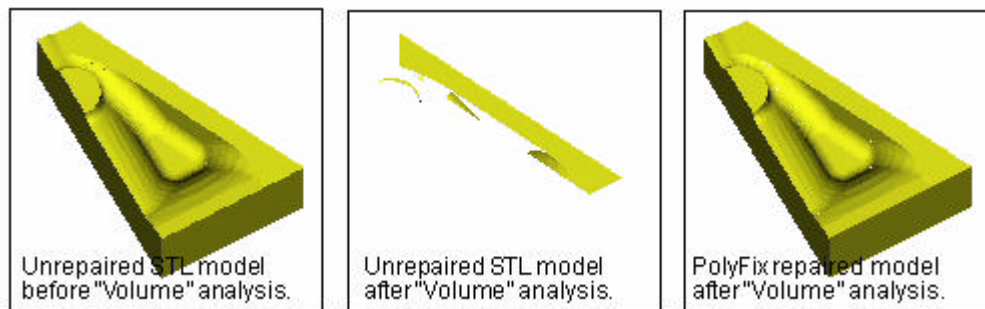
Go to the CGTech Online Library

PolyFix (Converter)

Location: **File menu > Convert > Polyfix**

The PolyFix Converter ensures the surface normals of Stereolithography (STL) and VERICUT Polygon Model files point uniformly outward from the model surface. (See how)

Model files with inconsistent normals often appear OK in VERICUT until the solid model database is established, such as occurs when cutting is started or the model analyzed. It is a one time effort to repair an inconsistent polygon model file. After repair, the model data will be processed much faster and more reliably.



Model files can be repaired interactively in VERICUT, via batch processing, or via running the PolyFix Converter externally ("polyfix" command file).

PolyFix Converter window features:

Input File—Name of the file containing the surface normals to repair.

Output File—Name of the file to receive repaired geometry data. When a model file containing multiple solids is processed, multiple output files are created—one per solid. Each file output has the base file name specified by the **Output File** field, with a unique number appended to it incremented by 1.

Input File Type—Type of input file to read. Options are: **VERICUT** or **STL**.

Output File Type—Same as **Input File Type**, except applicable to the output file.

Output File Format—Format of the output file. Options are: **BINARY** or **ASCII**.

Process—Processes (repairs) the model file data according to the current window settings. Repaired geometry is written to the **Output file**. Error and informational messages from the process are sent to the PolyFix Converter window message area, and to a log file named "polyfix.log" created in the working directory.

Clear—Clears the PolyFix Converter window message area.

What do you want to do?

See how to repair surface normals

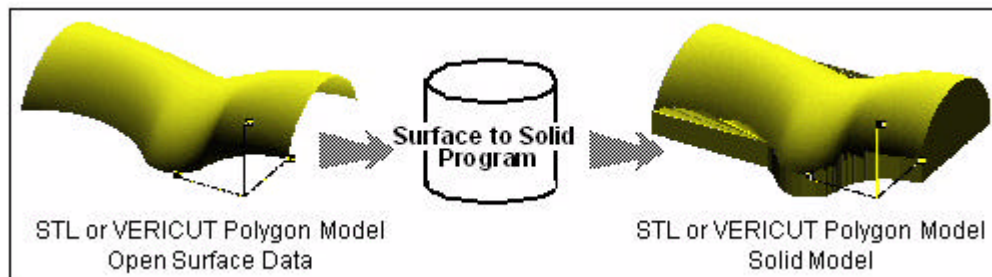
Go to the CGTech Online Library

Surface to Solid (Converter)

Location: **File menu > Convert > Surface to Solid**

The Surface to Solid Converter creates solid models from open surface STL and VERICUT Polygon Model files. (See how)

Parent surface(s) can be offset, then projected to a plane to create a solid model. This converter is particularly useful to create casting/forging stock models from design surface data.



The Surface to Solid Converter has the ability to output data as VERICUT or STL model files, in ASCII or binary file format. Solid model files can be created from surfaces interactively in VERICUT, via batch processing, or via running the Surface to Solid Converter externally ("sur2stk" command file).

Surface to Solid window features:

Input File—Name of the file containing the surface data to create a solid from.

Output File—Name of the file to receive converted geometry.

Input File Type—Type of input file to read. Options are: **VERICUT** or **STL**.

Output File Type—Same as **Input File Type**, except applicable to the output file.

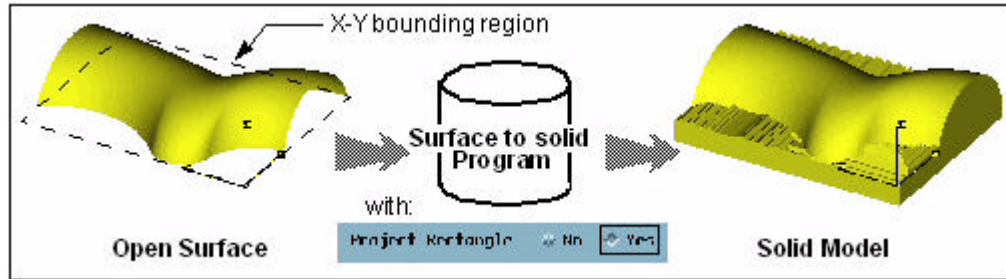
Output File Format—Format of the output file. Options are: **BINARY** or **ASCII**.

Tolerance—Specifies the tolerance used when offsetting and projecting the surface(s) to create a solid model.

Offset—Distance and direction in which to offset the design surface, as applied normal to the input surface. For example: ".100" applies an offset of .100 to the surface in the Z+ direction. A value like "-.100" applies the offset in the Z- direction, effectively "shrinking" the model.

Base Location—Specifies the Z value of the plane to become the base of the solid model. The surface is projected along the Z-axis to this plane.

Project Rectangle—When active, expands surface edges to create a rectangular base. The size of the rectangular base is determined by the X-Y bounding region of the surface data.



Transformation Matrix—Transforms the solid model. Use this feature is used when **Offset** and **Base Location** values were entered with respect to a tool path coordinate system instead of the model coordinate system.

Data entry format: dx,dy,dz,Ivx,Jvx,Kvx,Ivy,Jvy,Kvy

where “dx,dy,dz” specify the distance to the local coordinate system origin, and “Ivx,Jvx,Kvx,Ivy,Jvy,Kvy” specify the orientation of the local coordinate system X and Y axis vectors with respect to the surface model coordinate system.

Tip: Matrix values correspond to MSYS record values that appear in a UG CLS file.

Process—Processes (converts) the surface data according to the current window settings. Converted geometry is written to the **Output file**. Error and informational messages from the process are sent to the Surface to Solid Converter window message area.

Clear—Clears the Surface to Solid Converter window message area.

What do you want to do?

See how to convert surface data to solid models

Go to the CGTech Online Library

VDAFS (Converter)

Location: **File menu > Convert > VDAFS**

The VDA Converter converts geometry conforming to the VDA (Verband der Automobilindustrie Surface Interface) 2.0 specification to Stereolithography or VERICUT Polygon model files. (See how)

This converter is used to import complex castings, clamps, fixtures, and other design models into VERICUT, as well as generate design point data for use by the AUTO-DIFF function in VERICUT. VDA data can be converted interactively in VERICUT, via batch processing, or via running the VDA Converter externally ("vda" command file).

VDA Converter window features:

Input File—Name of the file containing the VDA data to convert.

Output File—Name of the file to receive converted geometry.

Output File Type—Type of output file to create. Options are: **VERICUT**, **STL** or **POINT** (Design Points file).

The VDA data elements that are converted depend on the type of output file being generated:

- POINT—single point -> Design Points file
- PSET—set of points -> Design Points file
- MDI—set of point/vectors -> Design Points file
- SURF—base surface -> VERICUT/STL model file
- FACE—trimmed surface -> VERICUT/STL model file

Output File Format—Format of the output file. Options are: **BINARY** or **ASCII**. Option not applicable to Point output file types.

Normal Direction—The direction in which surface normal vectors point, as viewed in the CAD system. Feature only applicable to VERICUT output file types. Options are: **OUTWARD**, **INWARD**, or **INCONSISTENT**.

Hint: When expecting to use VDA data for solid models in VERICUT, ensure that surface normals point unidirectionally outwards in the CAD model prior to outputting the VDA data.

Tolerance—Specifies the amount of chordal deviation allowed in 3-D space from the VDA surface when creating the converted surface. The converted surface is approximated using "facets", or triangles. Option not applicable to Point output file types.

Process—Processes (converts) the VDA data according to the current window settings. Converted geometry is written to the **Output file**. Error and informational messages from the process are sent to the VDA Converter window message area, and to a log file named "vda.log" created in the working directory.

Clear—Clears the VDA Converter window message area.

What do you want to do?

Help on VERICUT

See how to convert VDA data

Go to the CGTech Online Library

Binary CL (Converter)

Location: **File menu > Convert > Binary CL**

The Binary CL Converter converts binary APT CL data into ASCII low level APT tool path files. (See how)

Binary CL data is the pre-processed NC data typically sent to a post-processor. Binary CL data files can be converted interactively in VERICUT, via batch processing, or via running the Binary CL Converter externally ("bincl2v" command file).

Binary CL Converter window features:

Input File—Name of the file containing the binary CL data to convert.

Output File—Name of the file to receive converted APT tool path records.

APT Table File—Name of the APT Table file containing information for converting binary APT data into ASCII APT tool path records.

Input File Type—Type of binary input file to read. Options are:

- **IBM Mainframe APT**—APT/360, APT/370, APT/AC, or MDCAPT
- **Workstation IBM APT**
- **Workstation CATIA APT**
- **VAX APT**

Minimum Cutter Height—Default height used when cutter height is not defined in the binary APT data.

Output Circles—When active, outputs CIRCLE records for circular motion. Clearing this check box causes chordal GOTO points to be output representing each circle.

Process—Processes (converts) the binary CL data according to the current window settings. Converted tool path records are written to the **Output file**. Error and informational messages from the process are sent to the Binary CL Converter window message area.

Clear—Clears the Binary CL Converter window message area.

What do you want to do?

See how to convert binary APT data

Go to the CGTech Online Library

Mastercam (Converter)

Location: **File menu > Convert > Mastercam**

The Mastercam NCI Converter converts NCI files generated from Mastercam version 8 into ASCII low level APT tool path files. (See how)

Mastercam NCI data can be converted interactively in VERICUT, via batch processing, or via running the Mastercam Converter externally ("mcam2v" command file).

Mastercam Converter window features:

Input File—Name of the file containing the NCI data to convert.

Output File—Name of the file to receive converted APT tool path records.

Cutter Height—Height of cutting tools. This value is applied to all cutting tools in the NCI file.

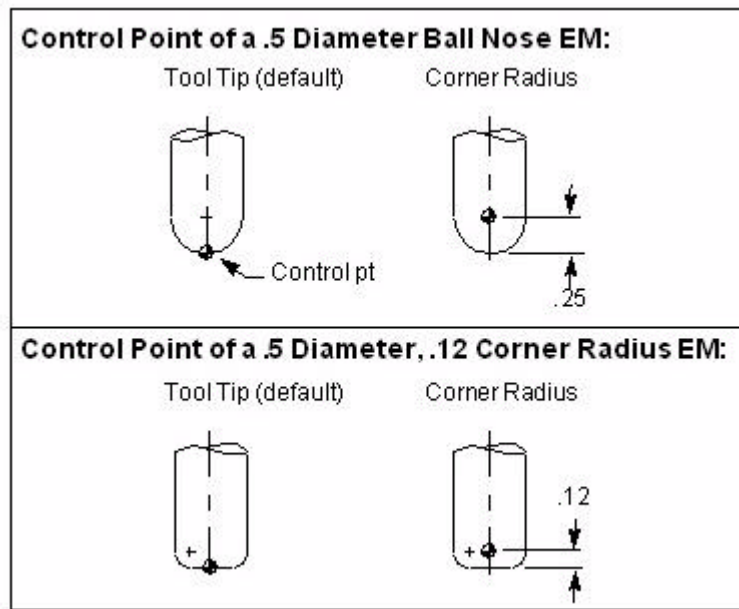
Circle Tolerance—Specifies the chordal tolerance (intol) for approximating circular motions with chordal GOTO points.

Tool Change Retract Height—Z value of the plane where the cutter is moved to for tool changes. The value entered is with respect to the tool path coordinate system.

Control Point—Describes the driven point of the cutter. Options are: **Tool Tip** or **Corner Radius**.

By default, VERICUT drives the tool tip (bottom center of the tool). Simulation of the material removed is based on motion commands and the tool shape relative to the control point. Examples follow.

Examples of setting the **Control Point**:



Process—Processes (converts) the NCI data according to the current window settings. Converted

Help on VERICUT

tool path records are written to the **Output file**. Error and informational messages from the process are sent to the Mastercam Converter window message area.

Clear—Clears the Mastercam Converter window message area.

What do you want to do?

See how to convert Mastercam NCI data

Go to the CGTech Online Library

Export Model

Location: **File menu > Export Model**

Opens a window to export the VERICUT model. Export formats include: Stereolithography (STL) or VERICUT Polygon model files, or IGES data. A cut VERICUT model is exported with machine cuts intact. When done, a message informs you how many IGES entities, or STL/VERICUT Polygon model triangles were created.

Export Model window features:

File—Name of the file to receive exported model data.

File Type—Type of file to export. Options are: **VERICUT**, **STL**, or **IGES**. IGES files conform to the IGES 5.2 specification.

File Format—Format in which STL or VERICUT model data is exported. Options are: **ASCII** or **Binary**.

Model Coordinates—Coordinate system in which model data is exported. Options are: **Stock** (stock component origin) or **Active Coord. Sys.** (ref. local coordinate system- ref. **Setup menu > Coord. System**).

Units—Unit measurement system in which model data is exported. Options are: **Inch** or **Millimeter**.

IGES export features:

Process—Controls the method of processing VERICUT model geometry into IGES data. Options are:

- **Machined Features**—Group similar machined features into like IGES data entities. This option is recommended for cut models with distinct differences between machined features, for example: aerospace structural components, pocketed parts, parts with strengthening ribs/walls and stepped floors, etc.
- **Grid**—Analyze the cut model using U V grid logic and group machined features by grid region. The grid value specifies the number of grid squares in U and V space, spread over the model surface. VERICUT attempts to create one surface within in each grid, or as few surfaces as possible. Increasing this number increases the size of the output file, and increases IGES data accuracy. This option is recommended for cut models with smooth or transitioning surfaces, for example: mold and die cavities and patterns, blended or sculptured surfaces, etc.

Connect angle—Angle which, if exceeded, divides the machined surface into multiple IGES surfaces. The value entered is in degrees. Increasing this number decreases output file size and decreases IGES data accuracy.

When processing machined features this value only affects multi-axis cut surfaces, such as ruled surfaces and torus sweeps generated by motion with a changing tool axis. On grid-processed models **Connect angle** affects the output of all machined surfaces. Note that when you increase the **Grid** number (more grid sections), the **Connect angle** can also typically be increased since fewer features are considered for each grid.

NURBS Only—When active, outputs cut model data in NURBS data entities (recommended for IGES data destined for Pro/Engineer). Clear this checkbox to output IGES data in any entity

format.

Minimum Edge—When active, specifies the minimum length (in units) of a boundary or "trimming" curve segment. Trimming curve segments with length shorter than the specified value are grouped with larger ones to decrease the size and complexity of trimming curves. Clear this checkbox to output IGES trimming curves as formed and as precise as possible.

Tips on using Minimum Edge:

- Increasing this number decreases the size of the output file and decreases IGES data accuracy. Entering too large a value can cause misalignment between trimming curves.
- Clearing the checkbox can cause VERICUT to output IGES data too small for your CAD system to process.
- Approximately 3/4 of the output file size can be attributed to trimming curve data. Changing the **Minimum Edge** value can have a dramatic affect on output file size.

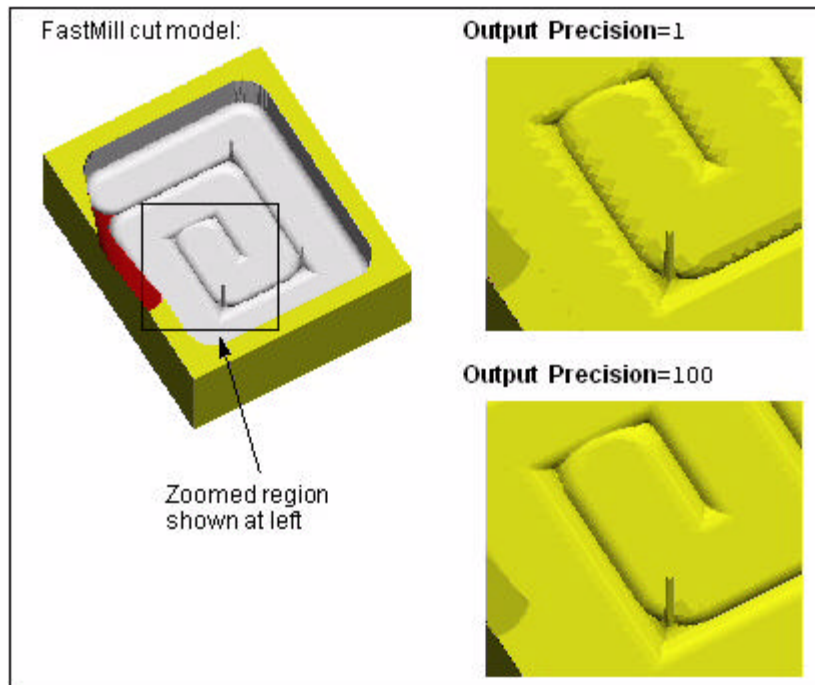
FastMill model export features:

Output Resolution—On FastMill cut models, this option controls when adjacent triangles are combined to reduce the output model size. **Output Precision** affects the quantity of triangles combined (see below). Options are:

- **Reduce Triangles**—Combine adjacent triangles.
- **As Displayed**—Do not combine triangles. Exported model accuracy is as seen on the screen.

Output Precision—On FastMill cut models with reduced triangles (see above), this feature affects the quantity of adjacent triangles that are combined. Moving the slide bar left combines more triangles to decrease output model file size and decreases IGES data accuracy. Moving the slide bar right combines fewer triangles to improve output model accuracy, however, output file size increases.

How **Output Precision** affects exported FastMill model quality:



Note: The **Display Resolution** property also affects exported FastMill model quality. (Ref. **File menu > Properties, General** tab.)

Output—Processes and exports the VERICUT model according to the window settings.

Tips on exporting models:

You can section cut models, then export what remains in the sectioned view. This can be helpful to export portions of complex models on computers with limited resources.

To export a higher quality model reduce the **Cutting Tolerance** (ref. **File menu > Properties: Tolerance** tab).

Toolpath File (Edit)

Location: **File menu** > **Edit** > **Toolpath File**

Toolbar short cut: 

Opens a text editor window with the current tool path file loaded and ready for editing. All standard text editing features are supported, such as: copy/cut, paste, search, etc.

Warning: Editing the tool path file can cause undesirable results, especially when edits are made to portions of the tool path which have already been processed by VERICUT. These edits can cause the tool path record number pointer to be out of sync with the edited tool path file. In these cases it is recommended to rewind the tool path file after editing, and restart the simulation with the last unedited record prior to the first addition/deletion.

Text File (Edit)

Location: **File menu** > **Edit** > **Text File**

Opens a text editor window-just load any ASCII text file and start editing. All standard text editing features are supported, such as: copy/cut, paste, search, etc.

Colors

Location: **File menu** > **Colors**

Toolbar short cut: 

Opens a window to define the colors seen in VERICUT. Colors are changeable at any time. The affect of changing colors is immediate.

Colors window features:

Define tab features

Assign tab features

Colors "Define" tab features

Features on this tab define colors for display in VERICUT, and control light brightness.

Shade Color list—List of colors that can be used to shade solid objects seen in VERICUT, such as: models, cutting tools, and machined cuts (maximum 12 colors). Once defined, assign the colors to solid objects using Assign tab features.

The number of shade colors also affects shade quality. The color band at the bottom of the window displays the shade variations of the selected color in the list, as impacted by the total number of defined colors.

Add, Delete—Adds (or deletes) colors in the list.

Note: The **Background** and **Foreground** colors should be different, or foreground-colored items may "blend in" with the background.

Red-Green-Blue sliders—Control the amount of red, green, and blue color components mixed to define a color. The exact value is shown in the field to the right of each slider (0=none to 255=maximum).

Color List—Opens a window of pre-mixed colors selectable for background, foreground, and shade colors.

Brightness—Controls the amount of light for shading solid objects. Options are: **High**, **Normal** and **Low**.

Flat Colors:

Background—Color of the graphics area background. Must be selected to change.

Note: You can also change the view attributes to display the background in a shaded color, or if its a machine view-display the ceiling, floor, and walls. (Ref. "" under **View menu** > **Attributes**)

Foreground—Color of items drawn on top of the VERICUT display, for example: coordinate system axes, lines indicating the bounding region of selected models/components, etc. Must be

selected to change.

Colors "Assign" tab features

Features on this tab assign colors to the cutting tool and machine cuts made in the workpiece.

Error—Color for highlighting unsafe machining conditions (default: "Red"). The "error color" will help you identify the following potentially dangerous situations:

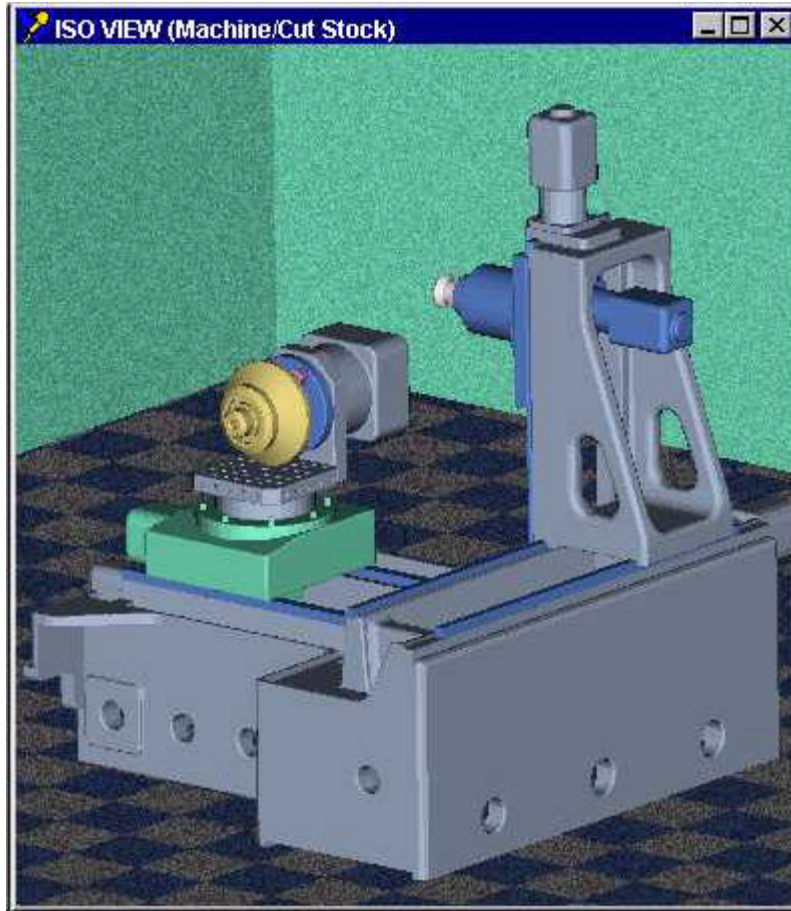
- Material removed while cutting too fast (known as a "fast feed" error)
- Cuts in clamps, fixtures, bolts, and other hardware holding the workpiece
- Material removed by non-cutting portions of the tool, such as: tool holders, extensions, etc.
- Gouges in a design model (via AUTO-DIFF)

Tool—Color of the tool when the cutting feed rate is safe (less than the **Fast Feed** threshold). In addition to shade color choices, a **Cut Color** option relates tool color to the current cut color (see "Cut Seq. / Color list" below).

Reset Cut—Color applied to all cuts in the workpiece when a **Reset Cut Color** operation is performed (see below). By default this color is set the same color as the stock model to give the appearance of uncut stock, however, any defined color can be selected.

Floor, Ceiling, Wall—Colors of the floor, ceiling and walls displayed in a machine view when the **Background** view attribute is set to display them. (Ref. **View menu** > **Attributes**)

A sample machine is shown below with the floor, ceiling and walls in the background. The tile affect on the floor is created by alternating the **Floor** and **Ceiling** colors.



Reset Cut Color—Resets the color of all cuts in the workpiece to be the **Reset Cut** color (see above).

Repaint—Repaints solid object colors based on current window settings. The new cut colors are determined by the current **Cut Seq. / Color list** (see below). Once repainted, the original cut colors can not be restored.

Cut Seq. / Color list—List of colors used for material cut with a safe cutting feed rate (less than the **Fast Feed** threshold). The cut color list associates colors with cut sequences in the tool path file. A new cut sequence is assumed when a tool with a different shape is loaded for cutting. All portions of the cutting tool shape are considered for differences.

Notes:

- When the number of cut sequences exceeds the number of cut colors, **Recycle Cut Color** determines the cut color for the remaining tools (see below).
- You can store cut colors with a tool's properties in a Tool Library. Set **Cut Color Method** accordingly to use the tool color property instead of the assigned cut color. (Ref. **Setup menu > Motion**).

Add, Delete—Adds (or deletes) cut colors in the list.

Recycle Cut Color—When active, recycles colors in the cut color list when the tool path contains more tools than cut colors defined. For example, assume that 4 cut colors are defined. When tool 5 is encountered VERICUT recycles cut color 1, tool 6 uses cut color 2, and so on.

Help on VERICUT

When the **Recycle Cut Color** checkbox is cleared, the last defined cut color is used for cut sequences that follow the last defined cut sequence color. Using the example above, tool 4, 5 and 6 would receive the cut color defined for cut sequence 4.

Properties

Location: **File menu** > **Properties**

Opens a window to set properties for the verification session, such as: cut mode, resolution and tolerance values that affect cut model display, accuracy, and motion simulation.

Properties window features:

General tab features

Tolerance tab features

Results tab features

Properties "General" tab features

Cut Mode—Mode in which VERICUT cuts the workpiece. The active choice determines if animated cutting motion is seen in the graphics area, and dramatically affects the time required to simulate cutting the workpiece. Availability and accuracy of some VERICUT functions are also affected. The cut mode can not be changed after cutting is started. In either mode, the **Cutting Tolerance** controls accuracy of the cuts in the model (ref. Properties "Tolerance" tab features below).

Options are:

- **Standard**—"Standard cut mode" simulates each cut performed by each tool path record and stores all information about the machining process. Cutting appears animated in the graphics area. This option is recommended when the highest degree of accuracy must be maintained.
- **FastMill**—"FastMill cut mode" processes all tool paths as quickly as possible without displaying animated cutting, then displays the final machined workpiece. VERICUT stores less information about the machining process than with the standard cut mode. This option is recommended for large tool paths with many small tool motions, such as typical automotive or mold and die-type parts.

FastMill exceptions:

- Intended for 3-axis through multi-axis milling on the Z+ side of the workpiece, where no undercuts are generated. All cuts must be visible when looking from the Z+ side toward the workpiece.
- One stock model. (If multiple Stock models are loaded, all but the first defined Stock model are automatically removed.)
- Fixture models are not used. (If loaded, they are automatically removed.)
- Tool holder shapes must be larger than the cutter, and have increasingly larger diameter dimensions, as defined from the cutter upwards.
- Can not use: a machine view, OptiPath tool path optimization, **Delete Material**, or AUTO-DIFF **Solid** or **Point** method (**Surface** okay).

Toolbar short cuts for cut modes and display resolution:



Display Resolution—Controls the quality of cut model display. Increasing resolution displays a more accurate representation of the cut model, and requires more computer resources and processing time to display and manipulate the model. Options are:

- **Standard**—A display resolution which allows most models to be manipulated with acceptable performance and display quality.
- **Auto**—Automatically sets display resolution based on viewing distance from the model. Resolution automatically increases when you zoom closer to see details on the model, and decreases as you zoom out.

Note: A coarse display resolution is automatically used when dynamically manipulating the model, such as rotation, zooming, etc. When the dynamic action is finished, the display is upgraded to the previous resolution.

Default Units—Specifies the default units (**Inch** or **Millimeter**) for VERICUT files opened that do not contain units identifiers, such as pre-4.0 VERICUT files.

Stock Consistency Check—When set prior to cutting, this option checks consistency of stock models loaded for cutting. VERICUT automatically makes the following repairs as needed:

- Check for "watertight" solid- repair improperly trimmed surfaces (overlaps and gaps), reconstruct insignificant missing surfaces.
- Check for outward pointing normals- reverse inward pointing normals to point consistently outward.

Tip: This option is highly recommended for stock models creating from importing IGES data.

Properties "Tolerance" tab features

Circle Radial Tolerance—Tolerance used when calculating radial distances from circle centers to their start and end points. Circle calculations that produce differences larger than this amount cause an "invalid circle" error, and the circle is not simulated.

Interpolation Tolerance—Tolerance used to interpolate NURBS motions, and machine positions during circle/arc and helical motions. This tolerance is also used by the **Detect 4-Axis Rotary Motion** feature to determine if tool positions lie in a 4-axis positioning motion- ref. **Setup menu > APT Settings: Rotary** tab.

Minimum Error Volume—Volume of material that, if removed, causes VERICUT to log an error and shade cut material using the Error color (typically red). Enter "0" to have VERICUT automatically calculate the volume amount, or enter volume in cubic units: inch or millimeter, depending on the system units.

Use **Minimum Error Volume** to make VERICUT more (or less) sensitive to the types of errors listed below.

- cutting with excessive feed rate
- cutting into clamps or holding fixtures
- non-cutting portions of the tool assembly removing material, such as shanks/holders.

Help on VERICUT

- removing material from Design models or previous Stock models protected by AUTO-DIFF

Error messages are sent to the VERICUT window message area and the Log file identifying the error type, tool path record responsible, and volume of material removed in error.

Model Tolerance—Tolerance used for creating solid models from revolved shapes, such as cylinders, cones, and turning models created from IGES profiles. Roundness of cylindrical tools/holders displayed in a machine view are also affected by this value.

Cutting Tolerance—Tolerance used for cuts applied to the workpiece. This value is similar to specifying an INTOL/OUTTOL when generating a tool path. Increasing this number increases the speed of the simulation, and requires less system resources and space to store the model database. However, the accuracy of cuts in the model is also reduced. The cutting tolerance also directly affects the results from AUTO-DIFF, OptiPath, holder collisions, Fast Feed errors, dynamic rotation speed, zoom speed, etc. In general, the value entered should reflect the accuracy needed for detecting cut-related errors and analyzing machined model features.

Tip: Best results are obtained when VERICUT is fully configured to simulate cutting the entire part before setting the **Cutting Tolerance**, e.g. all tool paths, all cutting tools that will be used, etc.

Properties "Results" tab features

Results file—Name of the Results file to receive overview information about the VERICUT session, including: current tool path and User file names, part images, machining times, optimization results, and links to the Log file and Optimized tool path file (if any). Results files are created automatically upon clicking **Info menu > Results File**.

Note: When a pre-5.0 User file is loaded, VERICUT automatically sets the **Results file** name based on the User file name.

Print

Location: **File menu > Print**

Toolbar shortcut: 

Opens a window to print images in VERICUT view ports. Each view port is automatically fit to the printable area and printed on a separate page. All view ports are printed.

Features on this window are standard print window features that enable you to access printer settings, specify number of copies, etc. Consult your printer documentation for more information.

Record

Location: **File menu > Images > Record**

Toolbar short cut for recording and playing back VERICUT images: 

Opens a window to record images in a VERICUT Image file. Images in all view ports are recorded. Images can be recorded interactively in VERICUT, or via batch processing. After recording is stopped, you can view the Image file via the **File menu > Images > Playback** function.

Image Record window features:

Snapshot Events—These options control when a snapshot image is recorded. Each selected event causes a snapshot to be recorded when that event occurs.

Options are: **Error**, **Cutter Change**, **N Records**, and **Processing Complete**.

Animation Control—These options determine when animation images are recorded, and how they will be replayed.

Options are: **Disabled** (no animation recorded), **Forward Only**, and **Forward & Backward**.

Filename—Name of the file to receive recorded images.

Append—When active, appends recorded images to the end of an existing Image file.

Append Delay—When appending images, inserts the specified number of seconds of delay prior to appending new images.

Insert Stop—Inserts a stop point into the Image file. The stop point is used by the **Stop at Inserted Stop** play back feature.

Insert Delay—Inserts the specified seconds of delay into the Image file.

Record—Starts recording animation images. Animation and snapshot images are recorded based on current window settings, window size, shape, and layout.

Stop—Stops recording animation images. Recording must be stopped before Image files can be viewed using **File menu > Images > Playback**.

Snapshot—Records a single snapshot image.

Hints:

The size of the graphics area directly affects the display size of the recorded Image file. Make the VERICUT window larger to record larger images.

Recording animations captures all changes in the graphics area. Recorded tool paths with lots of small motions can generate large Image files sizes. Consider using **Animation Speed** to reduce Image file size- ref. **Setup menu > Motion**.

Playback

Location: **File menu > Images > Playback**

Toolbar short cut for recording and playing back VERICUT images: 

Opens a window to load and play back an Image file recorded by the **File menu > Images > Record** function. Images are played in a separate "image player" window, enabling simultaneous access to the VERICUT main window and its functions. The size and configuration of the image player window is as recorded. However, this window can be moved, or minimized using standard window controls. Recorded machining status is also seen when the Status window is open while playing.

Image Playback window features:

Configure:

Speed—Controls how fast images are played back. Actual speed varies, depending on computer hardware. Stop the play back to access this feature.

Snapshot Delay—Number of seconds to pause after displaying a snapshot.

Image file—Name of the Image file to play.

Load—Loads the specified Image file into the image player.

Playback control (VCR) icons—Also known as playback VCR controls, these icons control playing and rewinding recorded images. To see what is associated with an icon, simply position the cursor over the icon and a tip appears.

Playback control options:

Icon:	Name:	Function:
	Play forward	Play images in the forward direction
	Play forward to next snapshot	Fast-forward until a snapshot is encountered
	Play forward to end	Fast- forward until file end is reached
	Stop	Stop playing back images
	Play backward	Play images in the backward direction
	Rewind to previous snapshot	Rewind to the previous snapshot
	Rewind to Beginning	Rewind to the Image file beginning

Control—These options provide additional stop control by event, or the ability to play in a

Help on VERICUT

continuous loop (without stopping).

Options are: **Stop at Error**, **Stop at File End**, **Stop at Cutter Change**, **Stop at Every N Events**, **Stop at Inserted Stop**, and **Loop**.

Done—Done playing, closes the image player window.

ShadeCopy

Location: **File menu** > **Images** > **ShadeCopy**

Opens a window to capture VERICUT images and format them for printing or importing into desktop publishing applications. ShadeCopy can also output printer-formatted files that you can move to other computers or printing devices. Images can be captured and output interactively or automatically (via **AutoSave**), or via batch processing.

Hint: Avoid having to reconfigure this window by creating a master User file containing the modified ShadeCopy values.

ShadeCopy window features:

Attributes tab features

Postscript Settings tab features

Output—Captures the current VERICUT image, formats the graphical data based on the window configuration, and outputs the image data as directed by the output destination.

When configuring this window for future use or batch processing, press **OK** (instead of **Output**) to save the window configuration without outputting a ShadeCopy image file.

ShadeCopy "Attributes" tab features

Features on this tab provide the attributes for formatting captured images.

Type—Type of format in which to capture VERICUT images. Printer-ready and electronic formats are available. Options are: **JPEG**, **PostScript**, **EPSF** (Encapsulated PostScript File), **TIFF** (Tag Image Format File).

Note: JPEG image support based in part on the work of the Independent JPEG Group.

Method—Method of coloring the captured image. Options are: **Standard** (black and white only), **Grayscale** (black and white with shading), **RGB Color** (Red-Green-Blue values for each pixel), **Palette Color** (Red-Green-Blue color tables), **CMYK Color** (Cyan-Magenta-Yellow-Black color tables).

Shading By—Method of shading black and white images. Options are: **Color** (shade by color differences and light intensity) or **Intensity** (shade by light intensity only, for example: the brightest yellow color is the same shade of gray as the brightest blue color).

Orientation—Controls image orientation on the printed page. Options are: **Landscape** or **Portrait**.

Output File—Name of the file to receive image data when the **Output** button is pressed.

Note: When ShadeCopy images are output via **AutoSave**, the file names in that window are used instead- see **File menu** > **AutoSave** for details.

ShadeCopy "Postscript Settings" tab features

Help on VERICUT

Features on this tab provide information about centering and annotating PostScript format images on the printed page.

Center / Margins options—Controls image placement on the printed page. Options are: **Center** (center image in the middle of the page) or **Margins** (place image according to specified margin dimensions-see below).

Left / Right / Top / Bottom—Margin dimensions, as measured from the corresponding edges of the paper.

Paper Size—Size of paper to print.

Annotation—When active, includes annotated text with the image. Annotation includes the current date, tool path file name, tool path record number being processed, and user specified text.

AutoSave

Location: **File menu** > **AutoSave**

Toolbar short cut: 

Opens a window to configure VERICUT to automatically save In Process files or ShadeCopy image files, such as JPEG, PS, EPSF, or TIFF files. Saving is based on user specified events, or when VERICUT detects errors during tool path processing. Auto-saving is especially useful to save data during batch processing.

AutoSave window features:

In Process tab features

ShadeCopy tab features

AutoSave "In Process" tab features

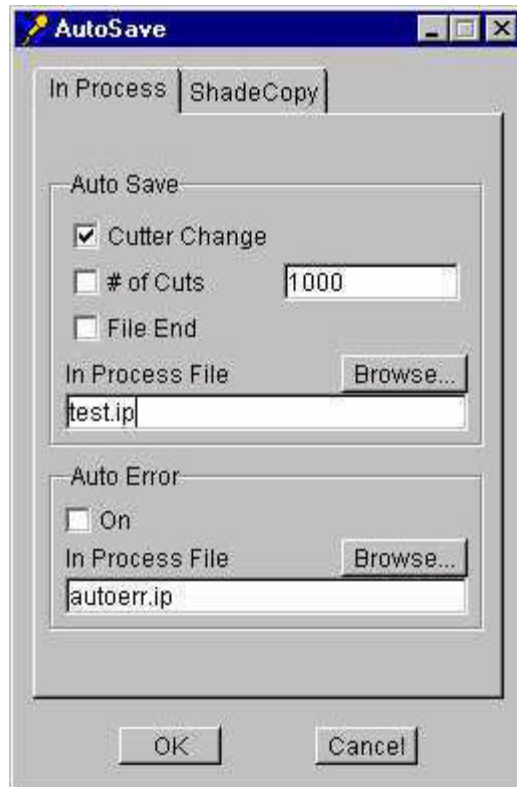
Features on this tab control automatic saving of In Process files.

Auto Save options—These options control when VERICUT automatically saves IP files. Each selected event causes an IP file to be saved when that event occurs.

Options are: **Cutter Change**, **# of Cuts**, or **File End**.

In Process File—Specifies the base name for IP files saved. Unique file names are maintained by appending the tool path number and line number of the record causing the event to the end of the base file name.

The naming convention: <IPfilename>n1_n2.ip is used for auto-saving files, where "IPfilename" is the name specified in the **In Process File** field, "n1"=sequential tool path number, and "n2"=number of the record causing the auto-save event.



The following examples assume the AutoSave window configuration shown above.

- Example 1: One tool path processed, tool change on record #10 => test1_10.ip
- Example 2: Two tool paths processed, tool change on record #10 in both tool paths => test1_10.ip (from tool change in tool path #1), test2_10.ip (from tool change in tool path #2)
- Example 3: Two tool paths processed, tool path one tool change on record #10, tool path two tool change on record #5 => test1_10.ip (from tool change in tool path #1), test2_5.ip (from tool change in tool path #2)

Auto Error On—When active, VERICUT automatically saves an IP file when an error is detected.

In Process File—Specifies the base name for IP files saved due to errors. Similar to **In Process File** for the **Auto Save** options (see above), unique file names are maintained by appending the line number of the record causing the error to the end of the specified IP file name.

AutoSave "ShadeCopy" tab features

Features on this tab control automatic saving of ShadeCopy image files.

Print Command—When cleared, sends graphical data to the specified **ShadeCopy File**. When active, executes the print command that sends raw graphical image data to the printer.

By default, the “prshade” command executes the “prshade” command file containing operating system commands typically used to print images on your computer. If printing fails, correct the

Help on VERICUT

entry in the **Print Command** field, or use an ASCII text editor (NOT a word processor) to edit the “prshade” command file to have the proper print command for your computer/printer.

Properties—Opens the **File menu > Images > ShadeCopy** window with access to settings for file formatting and printing. See that section for details.

Auto Save options—These options control when VERICUT automatically saves ShadeCopy image files-same as described above for IP files.

Auto Error On—When active, VERICUT automatically saves a ShadeCopy image file when an error is detected-same as described above for IP files.

ShadeCopy File—Specifies the base name for ShadeCopy image files saved when the **Print Command** checkbox is cleared-same as **In Process File** described above for IP files.

Recent Files

Provides a list of recently opened User files. To open a file in the list, select the desired file.

List contents are stored in the VERICUT Preference file.

Exit

Location: **File menu > Exit**

Exits (quits) the VERICUT program. Any files that were opened during the VERICUT session, such as the Log file, Image file, etc. are automatically closed. You can also exit VERICUT by clicking "X" at the top right corner of the VERICUT main window.

Session information is not automatically saved by VERICUT upon exiting. It is the responsibility of the user to decide if session information should be saved before exiting.

What do you want to do?

- Save the VERICUT user configuration (no 3-D model) before exiting

- Save the VERICUT 3-D model and user configuration before exiting

- Learn about other VERICUT functions

View menu functions

Layout

Location: **View menu > Layout >**

The options in the **Layout** menu control how many views of the VERICUT model are seen, and how they are arranged. All views are contained within the VERICUT main window.

The size and orientation of the model in each view is individually controlled. One view is considered by VERICUT to be the "active view". A view is made active by clicking in it, and is designated by a colored window header. Many VERICUT functions are effective only in the active view, including most viewing functions.

View menu > Layout options:

Standard—Sets views of the VERICUT model to one of the standard view layout choices. By default standard view layouts are tiled, but you can change this using other functions in the **Layout** menu, or via standard window management features, such as dragging, resizing, etc.

Add View—Adds a view. You can add as many views as desired. What is seen in a view (workpiece, machine, etc.) is determined by the view attributes- ref. **View menu > Attributes**.

Shortcut: You can quickly change the attributes or orientation of a view by right-clicking in the view, and selecting from the displayed menu.

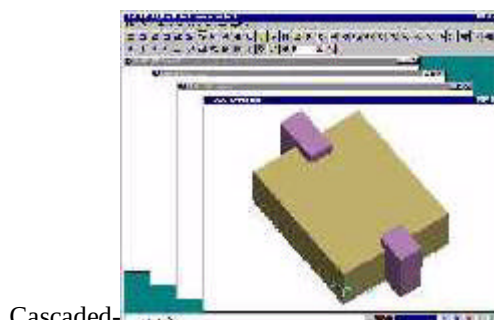
Delete View—Deletes a view. You can also delete a view via clicking "X" at the top right corner of the view window.

Cascade—Cascades views in an overlapping arrangement. Note that cascaded view layouts are not saved by VERICUT.

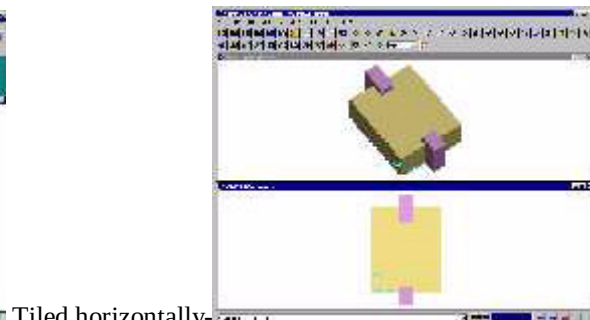
Tile Horizontally—Tiles views in a horizontal arrangement.

Tile Vertically—Tiles views in a vertical arrangement.

Example arrangements follow.



Cascaded-

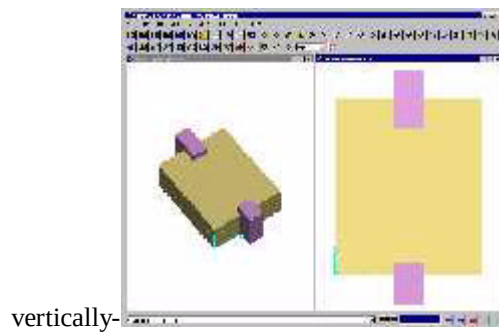


Tiled horizontally-



Tiled

Help on VERICUT



Orient

Location: **View menu > Orient**

Toolbar shortcuts: 

Opens a window to orient the view, including: rotate, zoom, pan, reverse, etc. You can view machining from any angle or distance, and change the view at any time during the simulation. Views of the workpiece, NC machine, or cutting tools seen in the Tool Manager are oriented in the same manner, as described by the sections that follow. In general, click in a view to make it active, then change view orientation as required.

Note: All view rotations are described with respect to the "view coordinate system"- see "About VERICUT views" in the *Using VERICUT* chapter for details.

View Orient window features:

Dynamic viewing options—These options use mouse actions to dynamically orient the object in view. Mouse actions are different, depending on the active option. See the table below for details. To see what is associated with an icon, simply position the cursor over the icon and a tip appears.

Dynamic viewing options (also available on the Toolbar):

Icon:	Name:	Action:
	Dynamic X Rotation	Horizontal rotation- drag mouse up/down in direction to rotate
	Dynamic Y Rotation	Vertical rotation- drag mouse left/right in direction to rotate
	Dynamic Z Rotation	Screen plane rotation- drag mouse left to rotate CCLW, right for to rotate CLW
	Dynamic XY Rotation	Horizontal/vertical combined rotation- drag mouse up/down/left/right in direction to rotate
	Dynamic Pan	Pan/translate- drag mouse in direction and distance to pan
	Dynamic Zoom	Zoom/magnify- drag mouse up to zoom in, down to zoom out

Shortcut: The following keys provide instant access from the keyboard to dynamic viewing options (press and hold keys while dragging): Dynamic Zoom—<Ctrl>, Dynamic Pan—<Shift>, Dynamic XY—<Ctrl> + <Shift>. Mouse actions are the same as above.

Standard drawing view buttons (XY, YX, YZ, ZY, ZX, XZ, V-ISO, H-ISO)—Orients the object in the selected standard drawing view and automatically fits the object in the view.

Rotation features:

Angles—Absolute rotation about the view coordinate system X Y Z axes, respectively. The three values are separated by spaces.

Increment—Degrees of incremental rotation to apply when a rotation button is pressed (see below).

Rotation buttons (X+/X-, Y+/Y-, Z+/Z-)—axis and direction in which to apply incremental rotation specified by the **Increment** value.

Static viewing options—These options use various mouse actions to zoom, fit, and reverse the object in view. See the table below for details. Static viewing options are also available on the Toolbar. To see what is associated with an icon, simply position the cursor over the icon and a tip appears.

Static viewing options:

Name	Icon	Action
Zoom to Box		Left-click and move mouse to trap area to zoom (press <Esc> to interrupt rubberbanding)
Zoom In / Zoom Out		Zooms in/out approx. 20% each time clicked
Fit		Fits object in active view
Fit All	none	Fits objects in all views
Refine (Display)		Refines the display to improve image quality
Reverse		Reverses view direction, as if you stood behind the object

Reset—Resets the view to the last orientation accepted by pressing **Apply**, or as it was when the Orient window opened.

Shortcut: You can quickly reverse or orient the model in one of the defined views by right-clicking in the view and selecting from the displayed menu.

Attributes (View)

Location: **View menu > Attributes**

Opens a window to control the attributes of a view, such as: what is seen in a view- workpiece or machine, light source, shading, etc. In general, click in a view to make it active, then change view attributes as required. Each view has its own view attributes.

View Attributes window features:

View Type—Controls what is seen in a view. Options are:

- **Workpiece**—Displays the stock workpiece and machining that occurs on it. This view supports full inspection capabilities on the machined part, including X-Caliper measurements and AUTO-DIFF model comparisons.
- **Machine**—Displays a 3-D NC machine, when defined. The uncut stock can be seen, however, material removal does not occur in this view type.
- **Machine/Cut Stock**—Same as above, except material removal does occur.

Note: In this VERICUT release, a workpiece cut in a **Machine/Cut Stock** view type does not support machined part inspection. Use the **Workpiece** view type for this activity.

Draw Mode—Controls how machine components are displayed. Options are: **Shade** (shaded solids), **Lines** (wireframe), **Hidden** (wireframe with back-facing lines hidden), **Mixed** (shade, lines, or hidden-as defined by each component), **Profile** (not used).

Toolbar short cut for shaded and wireframe draw mode: 

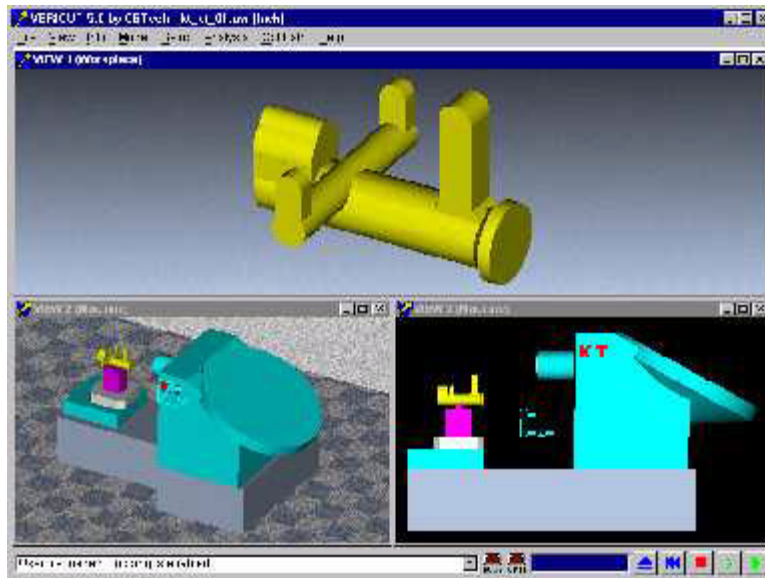
Light—I J K vector direction in which the light source points. The three values are separated by spaces, as described in the view coordinate system. A second light source shines on the opposite side of the model.

Shade Type—Method of shading cutting tool and model surfaces. Options are: **Gouraud** (smooth) or **Flat** (faceted) shading. Shade type does not affect the accuracy of the cut model.

Background—Controls the background seen in a view. Options are: **Flat** (monochromatic), **Shaded** (light-to-dark shading), **Walls** (walls, floor, and ceiling-machine views only). Colors for backgrounds are defined via the **File menu > Colors** function.

Sample backgrounds: Upper—**Shaded**, Lower left—**Walls**, Lower right—**Flat**

Help on VERICUT



Shortcut: You can quickly change the view type and background by right-clicking in the view, and selecting from the displayed menu.

Section

Location: **View menu > Section**

Toolbar short cut: 

Opens a window to section the VERICUT model. (See how) Sectioning removes the forward visual portion of the model, even when rotating the model. You can continue cutting in the sectioned view, or restore the original solid.

Reversing a sectioned view displays the portion of the model that was originally removed by sectioning. All active section planes are maintained. In other words, the model is reversed, then re-sectioned from the reverse side. An example follows.



Section window features:

Section on mouse pick—Sections through a point on the model indicated by a mouse pick. In "mouse pick mode", clicking off the model (in the background) restores the full model. Section plane orientation is controlled by the active section plane (**Plane 1-3**) and its **Plane Type** (see below).

Plane 1-3 —When **Plane Type** is set to something other than **Off**, the plane becomes an active section plane as defined using the data supplied below the heading. You can use any combination of one, two, or three individually controlled section planes.

Plane Type—Controls when a section plane is active, and its orientation. Section plane orientation and distance is described with respect to the current tool path coordinate system. **X**, **Y**, and **Z** define a section plane perpendicular to the specified axis. Using **Custom**, you can define a section plane at any orientation using **Normal I J K** and **Distance** (see below).

Distance—Distance along the **Normal I J K** vector to establish the section plane. The value can be entered, or supplied via mouse pick. A positive value is measured along the specified normal vector, negative is measured in the reverse direction.

Normal I J K—Specifies the I J K coefficients of a vector to section perpendicular to. These values are automatically set when **Plane Type**=**X**, **Y**, and **Z**, but are accessible with the **Custom** plane type. Values entered do not have to be normalized (vector length=1)-any number value can be entered.

Automatic sectioning features:

This group of features display a series of animated sectioned views.

Plane option list—Identifies the plane (Plane 1, 2, or 3) which sets the orientation for automatic sectioning.

Start, End—Similar to **Distance**, these values specify the start and end distances for the first and last planes for automatic sectioning. Values can be entered, or supplied via mouse pick.

Cuts—Quantity of cuts, equally spaced, between and including the start and end section planes.

Delay—Seconds of delay to pause between each section cut.

Auto—Begins the automatic sectioning sequence.

Restore—Restores the solid model from the sectioned view. When additional cutting has been performed in a sectioned view, the restored model has all machine cuts—even those applied to the sectioned portion of the model.

Notes:

Even though it has been sectioned away, the removed portion of the model is retained in memory. This enables you to measure the sectioned model as if the removed portion were still there.

Automatic sectioning is not compatible with **Section on mouse pick**.

Once started, automatic sectioning can not be interrupted.

Select/Store

Location: **View menu > Select/Store**

Opens a window to store and select custom views. (See how) Defined views are shown in the “View List”. There is no limit to the number of views you can store.

Select/Store View window features:

Add—Opens a window to specify the **View Name** of the view to add.

Modify—Opens a window to modify views in the View List. You can modify the view and its name.

Rename—Opens a window to rename views in the View List.

Delete—Opens a window to delete views from the View List.

Reset—Resets the view to what it was at the time of opening the Select/Store View window.

Shortcut: You can quickly reverse or orient the model in one of the defined views by right-clicking in the view and selecting from the displayed menu.

Axes (View)

Location: **View menu > Axes**

Opens a window to control when axes depicting various coordinate systems are displayed. Once displayed, axes symbols remain displayed until turned off via selecting the item in the menu again. Solid lines indicate axes are parallel or pointing out of the screen. Dashed lines indicate axes are pointing into the screen.

View Axes window features:

Component—When selected, displays **XcYcZc** axes that representing the coordinate system of a component. Each component has its own local coordinate system.

Model—When selected, displays **XmYmZm** axes that represent the coordinate system of a model. Each model has its own local coordinate system.

Machine Origin—When selected, displays **XmcYmcZmc** axes that represent the coordinate system in which an NC machine is defined.

Workpiece Origin—When selected, displays **XwpYwpZwp** axes that represent the coordinate system to which Stock, Fixture, and Design components are connected.

- *In a simulation where a machine IS NOT defined*, such as when processing APT-CLS tool path files, the workpiece coordinate origin is the origin of the non-moving "Base" component to which Stock, Fixture, and Design components are connected. In previous VERICUT versions, this was known as the "World coordinate system" (XwYwZw axes).
- *In a simulation where a machine IS defined*, such as when processing G-code tool path files, the workpiece coordinate origin is the origin the machine component responsible for carrying the Stock, Fixture, and Design components.

Coordinate System—When selected, displays **XcsYcsZcs** axes that represent the active coordinate system defined by the user, if any. If this options is selected and "cs" axes do not appear, then a user-defined coordinate system is not active. (Ref. **Setup menu > Coord. System**)

Set All, Clear All—When selected, turns on (or off) display for all axes symbols. Corresponding **Axes** options are all selected, or cleared, depending on the selected option.

Shortcut: You can change axes display by right-clicking in the view, and selecting from the displayed menu.

Notes:

Axes display status is stored in the Preferences file. Axes displayed when VERICUT is exited are displayed the next time you run VERICUT.

Axes are drawn in the **Foreground** color.

See also "VERICUT coordinate systems" in the *Using VERICUT* chapter.

Toolbar

Location: **View menu > Toolbar**

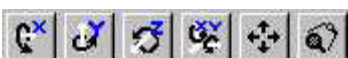
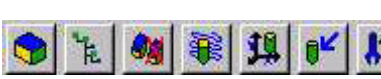
Opens a window to control when the Toolbar is displayed and the icons it contains. Toolbar icons provide quick and easy access to the most commonly-used VERICUT functions/features. The Toolbar can be closed, opened, or customized at any time. (See how)

View Toolbar window features:

Display Toolbar—When active, displays selected Toolbar groups on the VERICUT main window.

Set All, Clear All—Turns on (or off) display of all Toolbar icons. Corresponding options are all selected (or cleared).

View Toolbar group options:

Icons:	Toolbar name:	Functions:
	Close	Close Toolbar
	File	Open/save User and IP files, save/edit Toolpath file
	Options	Color, Image record/playback, print, AutoSave
	Display	Draw shaded/lines, section
	Dynamic View	Dynamic rotation-X, Y, Z, XY, pan, zoom
	Static View	Zoom, zoom in/out, fit, refine display, reverse
	Setup	Model def., component tree, delete matl., setup motion, coord. sys., G-code settings, Tool Mgr.
	Info	Info on tool path, status, log file
	Analysis	AUTO-DIFF compare, X-Caliper measure
	Cut Mode	OptiPath control, OptiPath Mgr., standard/FastMill cut modes, combined view, display res.

Refine Display—When selected, refines the display to be a high-quality image. Refining the display can useful to see model details after model manipulation, such as zooming, rotating, panning, etc. This action occurs automatically any time cutting is started.

Help on VERICUT

For information about VERICUT functions accessed by Toolbar icons, refer to the function names in the *Reference* chapter.

Look & Feel

Location: **View menu > Look & Feel**

Options that change the appearance of VERICUT's user interface. All windows are affected. The look and feel of VERICUT can be changed any time.

Options for user interface look and feel are: **Metal**, **Motif**, or **Windows**.

To change the user interface look and feel:

1. From the **View menu**, select **Look and Feel**.
2. Choose the option that provides the desired appearance.

Info menu functions

Toolpath (Info)

Location: **Info menu > Toolpath**

Toolbar short cut: 

Opens a window containing the tool path file being simulated. In this window, you can scroll through the entire file to see all the records, search for text forward and backward, print, and more. An arrow pointer indicates the current tool path record being processed. The window is updated when a new tool path file is processed, such as when a list of tool paths are simulated. (Ref. **Setup menu > Toolpath**).

Tips:

You can edit the tool path file using the **File menu > Edit > Toolpath File** function.

Status

Location: **Info menu** > **Status**

Toolbar short cut: 

Opens a window that provides status information about the simulation, and what would be occurring on the NC machine. Status information includes: tool path record being processed, machine and tool tip locations, information about the cutting tool, machining conditions, errors and warnings detected by VERICUT.

The Status window can be left open for monitoring, minimized, closed and opened at any time. Information seen is configurable, and is updated after each tool path record is processed.

SubSystem—When a machine with more than one subsystem is in use, this option controls which subsystem status to show.

Position:

Toolpath Record—Current tool path record being processed. Both the record text and sequential line number in the file are shown.

Machine Axes—Locations of machine axes resulting from processing G-Code tool path data.

Tool Tip—Current X Y Z tool tip location and I J K tool axis orientation with respect to the active tool path coordinate system.

Tool:

Change Record—Last tool change record processed. Both the record text and sequential line number in the file are shown.

Sequence—Sequential number of the last tool change event.

Number—Tool number in use (if specified). Examples follow.

Tool path type:	Sample tool chg record:	Status window "Number":
G-code data	T10M6	10
APT	CUTTER/...	none
APT	PPRINT/VERICUT-TC...	none
APT	LOADTL/1	1
UG CLS	LOAD/TOOL,4,...	4

ID—ID of a tool retrieved from a VERICUT Tool Library file, if any. A blank field indicates the current tool did not come from a tool library. “OVERRIDE” indicates that the current tool has been specified via the **Override** feature. If the ID differs from the **Number** field value described above, the tool was specified via a tool list.

Geometry—Cutter shape geometry. The method used to define the cutter shape is shown in square brackets “[]” followed by the description. “PROFILE” indicates an profile type of cutter

Help on VERICUT

definition.

Description—Description of a tool retrieved from a VERICUT Tool Library file. A blank field indicates a description for this tool was not defined in the library, or the tool did not come from a tool library.

Status:

Errors—Quantity of errors detected by VERICUT. A blank field indicates no errors detected. Consult the Log file for details about the errors.

Warnings—Similar to **Errors** above, except applies to warnings detected.

Time—Time anticipated to machine the part (as simulated by VERICUT). To calculate anticipated machining time, VERICUT records time for tool motions based on specified cutting feed rates and rapid rates.

OP Time—Time anticipated to machine the part with an optimized tool path file. This field is only updated when the current tool path is being optimized by OptiPath. To estimate machining time saved by OptiPath, compare this value against the **Time** field value.

Time %—Percentage of time in feed rate mode. This value can be helpful when judging tool path efficiency. For example, a high value suggests an efficient tool path, since a value less than 100 percent means that the tool is moving at rapid feed and is probably not removing material.

Distance—Tool movement distance in inches or millimeters, depending in the VERICUT session configuration.

Distance %—Percentage of tool movement distance in feed rate mode. Similar to the **Time** versus **Time%** relationship described above, this value can also be helpful when judging tool path efficiency.

Coolant—Current programmed coolant condition. A blank field indicates no records that control coolant that have been processed.

Feed & Speed:

Feedrate—Current programmed feed rate. A blank field indicates no records that control feed rate that have been processed.

OP Feedrate—Optimized feed rate, as calculated by OptiPath.

Spindle—Current programmed spindle speed. A blank field indicates no records that control spindle speed that have been processed.

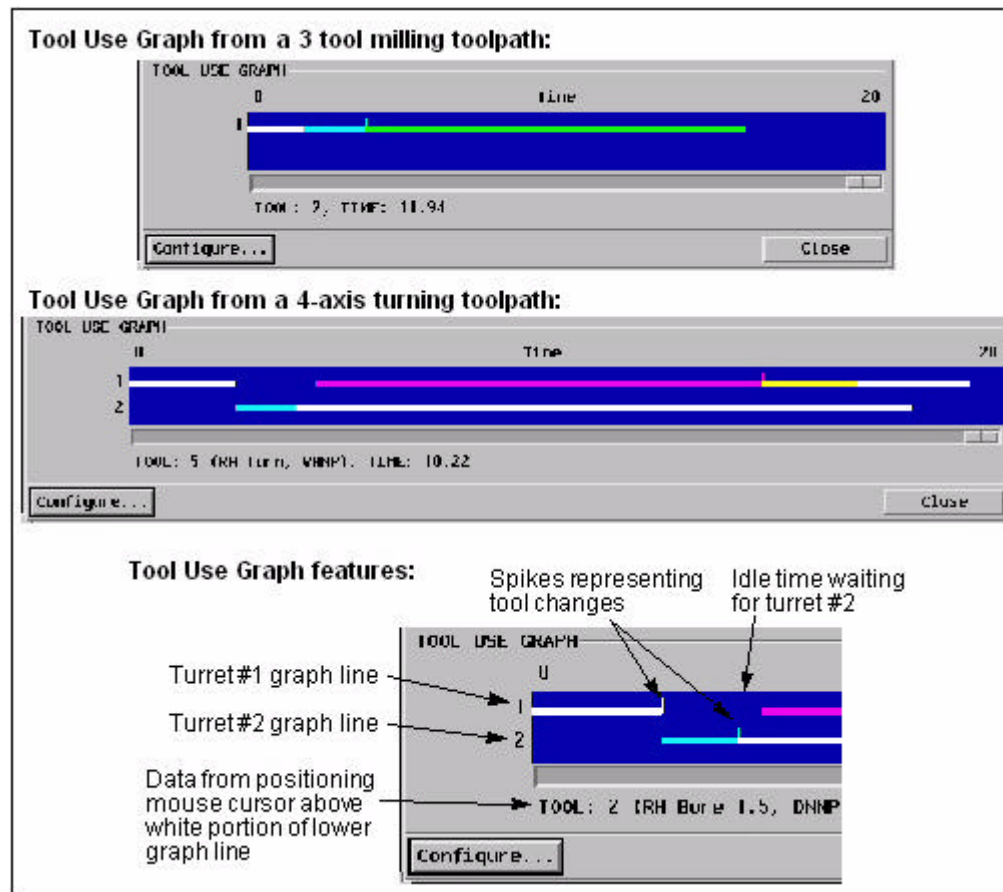
OP Spindle—Optimized spindle speed, as calculated by OptiPath.

Tool Use:

Tool Use graph—When active, reveals the Tool Use Graph—a graph that shows the machining time spent by each simulated tool. Each tool that cuts the part is shown in the graph with its cut color. Tool changes are indicated by a spike in the graph line. Positioning the mouse cursor on or above a graph line displays data about that tool.

A 4-axis turned part has 2 graph lines; one for each turret. A gap in a graph line indicates idle time spent waiting for a turret to synchronize. Positioning the mouse cursor above a gap in the graph displays the amount of idle time.

Sample Status window:



Configuring the Status window:

Configure—Opens a window to configure the Status window. Select items you want to display, clear items to remove from the Status window. You can reconfigure the window at any time.

- **Time Interval**—Specifies the length of time which is visible on the Tool Use Graph. A scroll bar provides access to unseen graph areas. The time interval can be changed at any time.

Tips:

Status is recorded by the **File menu > Images > Record** function, and can be seen in the Status window when playing back the Image file.

You can send status information to a text Status file. To accomplish this, set the CGTECH_STATUS environment variable to the name of the file before or when VERICUT is executed.

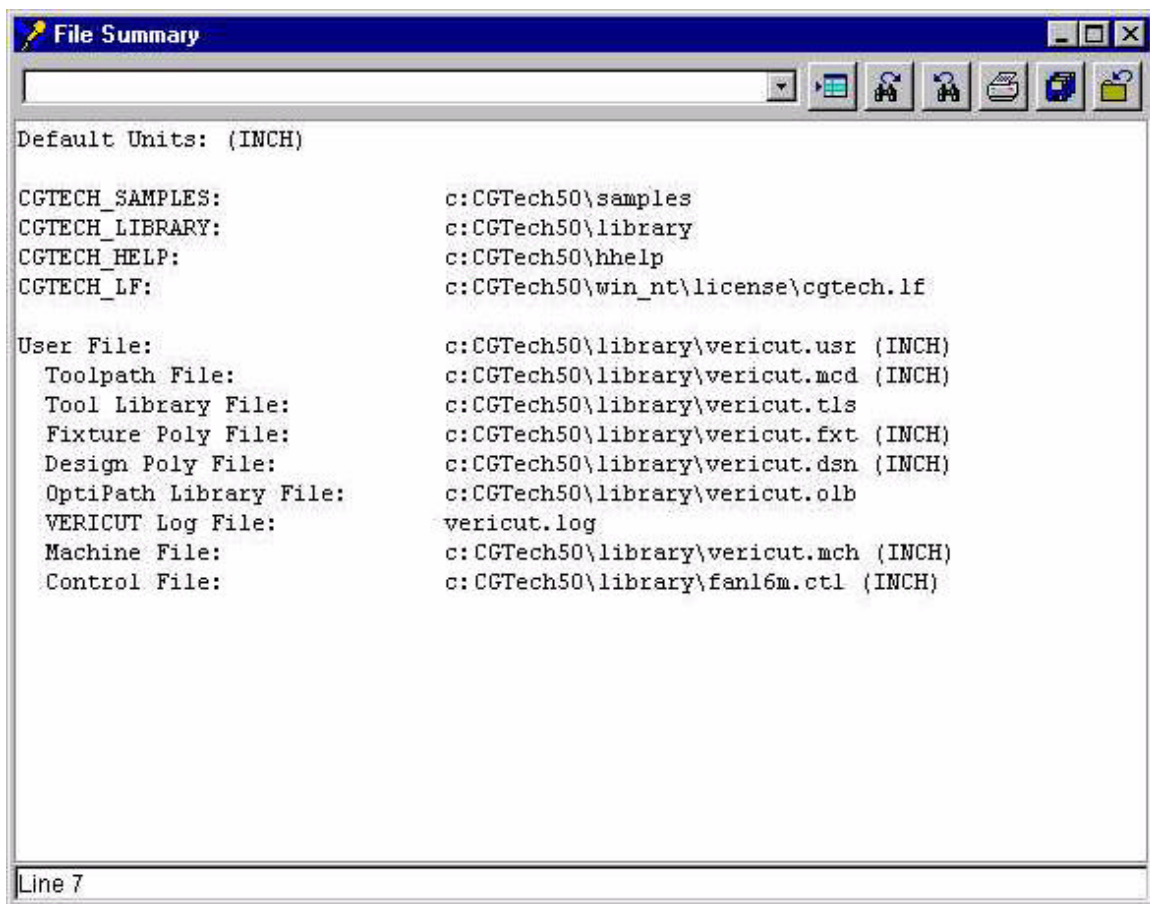
File Summary

Location: **Info menu > File Summary**

Opens a window reporting a summary of environment variables and files currently in use by VERICUT. All standard text viewing features are supported, such as: search forward/backward, go to line number, print, etc.

A "**Copy all files**" option  provides a quick and easy way to store a copy of the files in a new directory. This action is useful when preparing a set of files to send to CGTech, or other destination.

Sample File Summary window:



G-Code Report

Location: **Info menu > G-Code Report**

Creates a G-Code Report file containing information about how the current NC control configuration will interpret a G-code tool path file. Reported information includes: machine codes present in the tool path and macros (actions) they will call, variables used (if any), cutting tools used, subroutines defined and referenced, and more.

Control Report

Location: **Info menu > Control Report**

Creates a Control Report file containing information about how the current NC control configuration will interpret machine codes. Reported information includes all the G-codes the control is configured to handle and macros (actions) they will call, subroutines, and other supported control features.

VERICUT Log

Location: **Info menu** > **VERICUT Log**

Toolbar short cut: 

Opens a window containing the VERICUT Log file. This file contains session information, such as error, warning and informational messages about the verification session. In this window, you can view, print, change then name, or reset Log file contents.

Results File (Info)

Location: **Info menu > Results File**

Opens your browser with an HTML-formatted VERICUT Results file for review. This file contains overview information about the VERICUT session, such as: the current tool path and User files, part images, machining times, optimization results, and links to the error Log file and optimized tool path file (if any).

Exact contents of this vary, depending on what VERICUT was configured to do during the session. For example, if OptiPath was not turned on to optimize the tool path, then optimization results do not appear.

Model menu functions

Model Definition

Location: **Model menu > Model Definition**

Toolbar short cut: 

Opens a window to define or import the models that represent the solid objects used in the simulation. VERICUT uses "components" to describe the function of various solid objects, while models provide 3-D size and shape for material removal and collision checking. To learn more, see "About modeling in VERICUT" in the *Using VERICUT* chapter.

Modeling window features:

Active Component—Component active for editing. The active component is automatically set when you select items from the graphics area or Component Tree window. However, you can select a different component via picking it from the option list.

"Component Type" is also shown for reference. To change this value, use **Edit > Component Type** in the Component Tree window.

Selected—Identifies what is selected for editing: a **Model** or the **Component**. This option is automatically set when you select items from the graphics area or Component Tree window. However, you can change what is selected for editing via clicking the other option.

The remainder of the window appears different, depending on the tab selected. See the appropriate section below for specific information.

Model tab features

Position tab features

Assemble tab features

Component Attributes tab features

Import tab features

Tips:

Features used to manipulate components are displayed in a shortcut menu when a component is selected and the right mouse button is clicked in the Component Tree window. Similarly, features used to manipulate models are displayed when a model is selected and the right mouse button is clicked.

Click repeatedly on a model in the graphics area to toggle selection between the model and its component. Refer to the Component Tree window to review what is selected.

Using the center mouse button of a three button mouse, click repeatedly on a model in the graphics area to toggle selection between different models along the line of sight from the pick point. Use this technique to select models hidden from view by an object in the foreground.

Model Definition - Model tab

Toolbar short cut: 

Features on this tab are used to interactively define and add models to components. (See how) The models can be simple parametric shapes, or refer to external model files.

Type—Type of model being defined. The active choice configures the model description area below this option with the values needed to define the selected model. The table below lists the available model types and corresponding descriptions.

Model type:	Icon appearing in the component tree:	Description
Block	 Block	Length (X), Width (Y), Height (Z)
Cone	 Cone	Height (Z), Base Radius, Top Radius
Cylinder	 Cylinder	Height (Z), Radius
Model File (STL or VERICUT)	 STL  Vericut Polygon	Model File, Normals*, Units

* **Normals** option applies to STL file models—indicates the direction in which STL data normals point: **Outward**, **Inward**, or **Inconsistent**. The “Inconsistent” option should only be used after “Outward” and “Inward” have been attempted with unfavorable results, since it requires significantly more processing time and computer resources.

Auto Block—When defining a "Block" model, this feature automatically defines and locates a stock block based on tool positions in the current tool path file(s). Rapid and APT-CLS “FROM” tool positions are ignored during this calculation. The block size and location guarantee that the tool path will cut the block.

Offset—Offset apply to a block created using **Auto Block** (see above). The offset is applied in all dimensions.

Color—Color of the model. Available colors are determined by the colors defined on the **File menu > Colors: Define** tab. Models connected to the component have the option to “inherit” or override the component’s color.

Add—Adds the model defined by the current window values to the active component.

Model Definition - Position tab

Toolbar short cut: 

Features on this tab are used to move a component or model by translating or rotating. (See how)
Movement is relative to the active component origin.

Translate:

From / To / Move—These features translate the model. Translation is incremental, as calculated from the "**From**" point to the "**To**" point location. Incremental translation is applied each time **Move** is pressed.

XYZ values can be entered (separated by spaces), or filled in by clicking on a model. In general, graphical selection supports picking corner points and midpoints of model geometry. A crosshair and vector show you the pending pick-point location.

Tip: If the applied translation is incorrect, press **Undo** to return the object to its previous location.

Rotate:

Center of Rotation—Specifies XYZ center of rotation used when the rotation buttons are used (see below). XYZ values can be entered (separated by spaces), or filled in by clicking on a model, as described above for **From / To / Move**.



To see the center of rotation, press . To remove the center of rotation symbol press this button again, or close the Modeling window.

Increment—Specifies incremental degrees of rotation applied when one of the rotation buttons are pressed.

Rotation buttons—(X+/X-, Y+/Y-, Z+/Z-) Applies the incremental degrees of rotation specified in the **Increment** field. Rotation occurs about the **Center of Rotation** relative to the active component origin.

Tip: If the applied rotation is incorrect, press **Undo** to return the object to its previous location.

Location:

Relative Location—When active, object movement is relative to the parent component. Examples follow.

- *Positioning a component*—component is moved relative to the parent component origin.
- *Positioning a model*—model is moved relative to the component origin.

Machine Location—When active, object movement is relative to the machine origin.

Position—Specifies the absolute XYZ position of the object relative to the parent component when **Relative Location** is active, or relative to the machine origin when **Machine Location** is active. Values in this field are separated by spaces.

Angles—Specifies the absolute XYZ rotation of the object relative to the parent component when

Help on VERICUT

Relative Location is active, or relative to the machine origin when **Machine Location** is active. Values in this field are separated by spaces.

Undo—Returns the object to its previous location, or as it was when the Modeling window was opened.

Model Definition - Assemble tab

Toolbar short cut: 

Features on this tab are used to move a component or model by assembling it with others. These objects are assembled by mating or aligning one to three planar surfaces with surfaces on other models. (See how) If a non-planar surface is selected, VERICUT constructs a tangent plane at the pick point.

The relationship of surfaces being mated or aligned is known as a "constraint". Constraints must identify a unique location to move object to, or additional constraints must be defined.

These general steps define a constraint for assembly:

1. Choose the constraint type.
2. Select a surface on the object to move.
3. Select the surface to move the object relative to.
4. Press **Apply**.

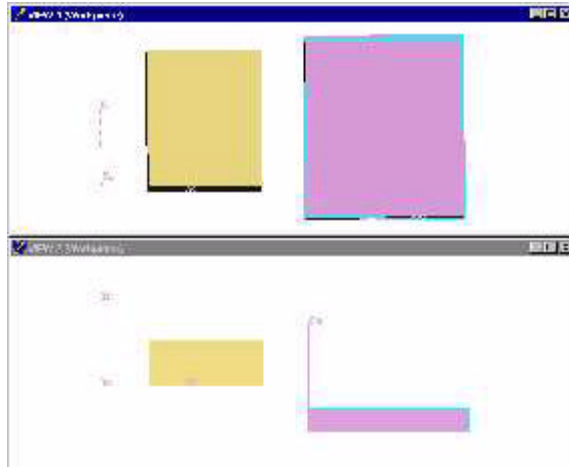
Window features:

Constraint Type—Specifies how selected surfaces are constrained (related) to cause object movement. After selecting two surfaces to satisfy a constraint, VERICUT highlights the constraint with a checkmark.

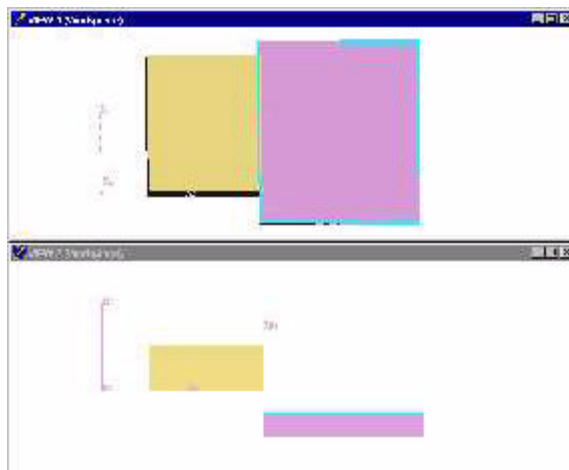
Constraint type options are:

- **Mate**—Moves the object so the selected surface opposes the surface selected on the second object (normals oppose each other). Mated surfaces are coplanar, may or may not be in contact with each other.
- **Align**—Moves the object so the selected surface is aligned with the surface selected on the second object (normals point in the same direction). Similar to mated surfaces, aligned surfaces are also coplanar. However, they may or may not be in contact with each other.

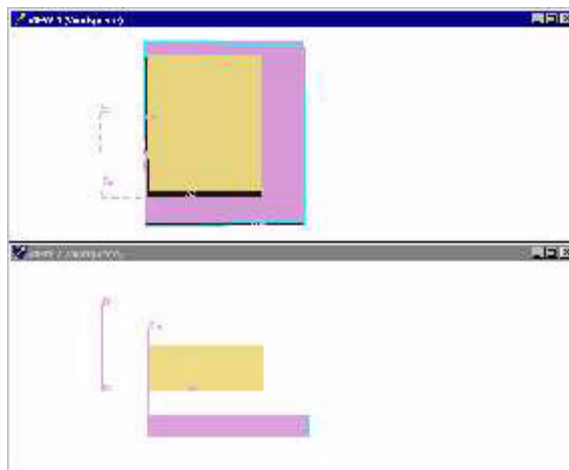
Consider the starting condition below for the examples that follow.



Mated surfaces- the left side of the magenta model is mated with the right side of the yellow model.



Aligned surfaces- the left side of the magenta model is aligned with the left side of the yellow model.



Offset—Distance and direction in which to offset constrained surfaces, as applied normal to the surface.

Help on VERICUT

Reset—Resets all constraints to receive new data.

Undo—Returns the object to its previous location, or as it was when the Modeling window was opened.

Model Definition - Component Attributes tab

Toolbar short cut: 

Features on this tab define the attributes of a component. (See how) VERICUT handles components and their solid models differently depending on the component's type and attributes. Most attributes apply to components used in building NC machines.

Color—Color of the component. Available colors are determined by the colors defined on the **File menu > Colors: Define** tab.

Visibility—Controls the views in which the component is visible. Options are:

- **Blanked**—Component is blanked from all views and cannot be seen.
- **Workpiece View**—Component is seen in the workpiece view.
- **Machine View**—Component is seen in the machine view.
- **Both**—Component is seen in both the workpiece and machine views.

Machine:

Motion Axis—Specifies the axis of motion for a linear, rotary or turret type component, relative to the component coordinate system. Use a “moving tool” philosophy to specify the motion axis and direction, even though the component may actually move on the machine instead of the tool.

- *Linear motion component*: the motion axis is the axis in which linear motion occurs.
- *Rotary and turret components*: the motion axis is the axis about which rotation occurs. The positive rotation direction is counterclockwise when viewed from the positive end of the motion axis towards the component origin.
- *Tool and spindle components*: the motion axis is the aligned with the tool axis, typically “Z”.

Disable Auto Direction—When active, disables VERICUT’s sense of the “moving tool philosophy” for that component. This feature is designed for components that are disconnected and reconnected in different orders on the machine. This causes the component motion direction to remain constant regardless of where it is connected.

Reverse Direction—When active, reverses the component motion direction.

Mixed Mode—Controls how the component is displayed when displayed in **Mixed** draw mode (ref. **View menu > Attributes: Draw Mode**).

Connect To—Component to which the active component is connected. The component being connected to is known as the “connect component”. Components must be connected in the same way as they are in the real world. Improper connections cause incorrect NC machine movement and simulation, especially when rotary components are involved.

Rapid Rate—Feed rate for motion-type components moving in rapid positioning mode (e.g. G0). For linear motion components the value entered is units/minute where “units” reflect the units of the tool path file. For rotary motion components the value entered is degrees/minute.

SubSystem—Subsystem which commands a linear or rotary motion type component to move.

Help on VERICUT

Any alpha-numeric sub-system name can be entered (may include spaces). Enter "*" (without the quotes) for motion axes included in all machine subsystems.

Subsystems are used in machines with multiple motion components that are commanded to move via the same G-code word, for example "X". By default, a machine uses one subsystem for all motion components, assuming they are commanded to move via unique G-code words.

When multiple motion components are driven by the same G-code word, such as can occur in 4-axis lathes or mills with multiple independently controlled heads, define different subsystems to control which component moves when the motion command is processed. Then, configure the control so that other codes in the tool path activate the appropriate subsystem for movement. A full set of tables are available to support each machine subsystem.

See "Sample-demo files" in the *VERICUT Files* chapter for examples of machines that use multiple subsystems.

Tool Index—Index number of a Tool component. When defining machines with multiple tool load positions, such as a multi-spindle mill or a turret lathe machine with tools loaded in the turret, define additional Tool components with different index numbers. Values entered are integer numbers.

Use the "ActiveTool" macros to activate each Tool component for tool changes. For example, call "ActiveTool2" to activate the Tool component with **Tool Index=2**, and so on.

Accel/Decel:

Acceleration / Deceleration—Velocity at which the NC machine component can accelerate (or decelerate). Values entered are units per second per second, depending on the machine units.

Max Feed Velocity—Maximum speed axes can travel while in feed mode. The value entered is in units per minute, depending on the machine units.

Max Velocity for Direction Change—Maximum feed rate for motions that cause an axis to decelerate to zero velocity, such as occurs when turning at a corner.

Model Definition - Import tab

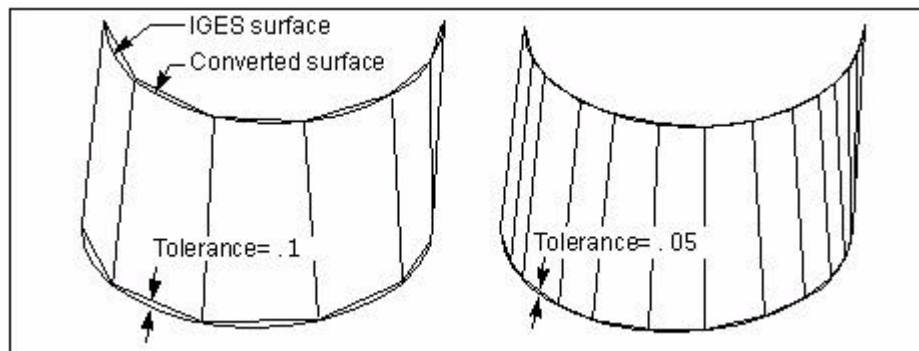
Toolbar short cut: 

Features on this tab import IGES model files for use as stock, fixture, design, or machine component models. (See how) The component type determines how the converted IGES model will be used in VERICUT. An option is also available to create a solid model from an IGES surface. The surface can be used as is, or offset before converting it into a solid.

During import, VERICUT converts the IGES model data into a VERICUT Polygon file (.ply extension) and automatically refers the converted file. This window functions similar to the IGES Converter window-ref. **File menu > Convert > IGES.**

IGES File—Name of the file containing the IGES data to import.

Tolerance—Specifies the amount of chordal deviation allowed in 3-D space from the IGES surface when creating the converted surface. The converted surface is approximated using "facets", or triangles.



Offset Surface:

Offset Surface—When active, creates a solid model from an open IGES surface.

Offset Amount—Distance and direction in which to offset the design surface, as applied normal to the input surface. For example: ".100" applies an offset of .100 to the surface in the Z+ direction. A value like "-.100" applies the offset in the Z- direction, effectively "shrinking" the model.

Base Location—Specifies the Z value of the plane to become the base of the solid model. The surface is projected along the Z-axis to this plane.

Extend to Rectangle—When active, expands surface edges to create a rectangular base. The size of the rectangular base is determined by the X-Y bounding region of the surface data.

Component Tree

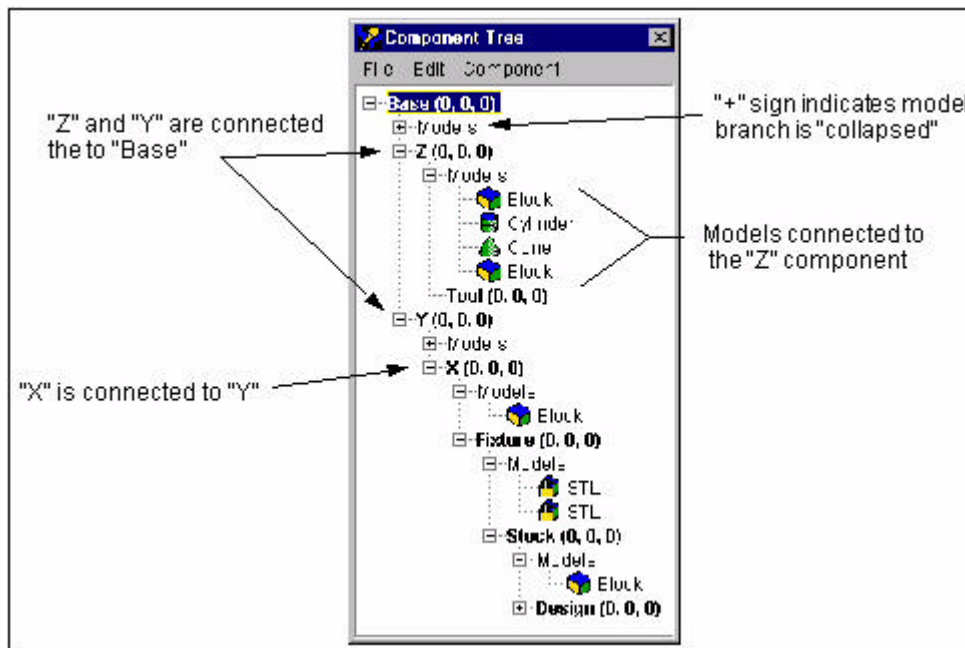
Location: **Model menu > Component Tree**

Toolbar short cut: 

Opens a window that shows the "tree" of components and how they are connected, as well as provides a fast and easy way of manipulating components and their models. (See how) The component tree shows defined components by name and their connect positions. Indentation shows how components are connected in the tree.

Select a component by picking its name in the tree. Actions performed using Components Tree window features are performed on the active (selected) component. To learn more about modeling and using the component tree, see "About modeling in VERICUT" in the *Using VERICUT* chapter.

Sample Component Tree window:



Note: Always **Reset** (press ) after changing components in the component tree.

Component Tree window features:

File menu—Functions in this menu open or save a Machine file containing data that describes NC machine construction, kinematics, and other properties.

Edit menu—Functions in this menu manipulate components in the tree, as well as specify the component name, type, and attributes.

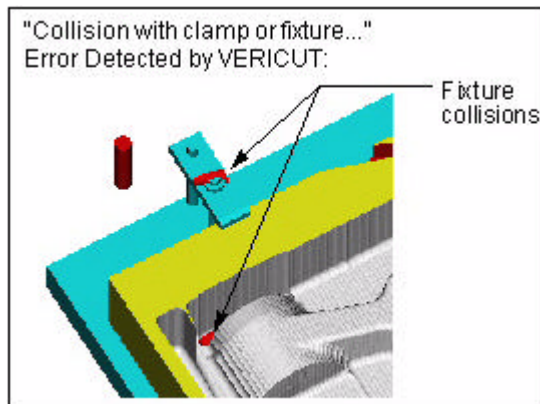
- **Cut / Copy / Paste / Delete**—Cuts, copies, pastes, or deletes a selected component in the component tree. All connected components and models are also affected.

- **Rename**—Allows you to rename the selected component. Type the new component name.
- **Component Attributes**—Accesses the features on the **Model menu > Model Definition: Component Attributes** tab-see that section for details.
- **Component Type**—Edits the component's "Type" setting which determines the component's function in the simulation.

Component menu—Functions in this menu insert or append new components to the selected component in the tree. The component type describes the component's function in the simulation. Choose the appropriate component type from the option list, based on the component's function in the simulation. Use **Edit > Component Attributes** to supply any other attributes that are needed.

Types of components:

- **Base**—A non-moving component to which other components are connected. Every machine definition begins with this component.
- **Stock**—Workpiece to be machined. This component type must be present in all simulations.
- **Fixture**—Hardware used to hold the workpiece for machining, such as: base plates, clamps, bolts, vises, pallets, etc. VERICUT assumes fixtures are not to be machined. Cuts in a fixture are shaded using the red **Error** color and an error is issued to the Log file.



- **Design**—Solid or surface representation of the theoretical designed part.
- **Design Point**—Point representation of the theoretical designed part.
- **X [Y,Z] Linear**—Linear motion axes on an NC machine.
- **A [B,C] Rotary**—Rotary motion axes on an NC machine. The component origin is the pivot point of rotation.
- **U [V,W] Linear**—Co-linear motion axes on an NC machine. Motion axes are parallel to X,Y,Z axes, respectively.
- **A [B,C] Turret**—Indexing tool turret that rotates about X, Y, or Z axes. Like rotary components, the turret component origin is the pivot point for rotary indexing. The presence of these components cause all connected tools to be displayed making it possible to detect collisions resulting from turret indexing.

A Turret component is commanded to move using a tool change command (e.g. Tn). The turret indexes based upon activating a Tool component connected to the turret. Tool component orientation determines the tool index position in the turret. Rotary control settings determine

how turrets rotate, e.g.: rotation direction, etc.

- **Spindle**—Spindle on an NC machine. The function of this component is different, depending on the machine type: milling or turning. Milling machine spindle- portion of the machine where cutting tools are loaded. Turning machine spindle- rotating component that turns the workpiece about the machine Z-axis.
- **Tool**—The location where cutting tools are loaded in an NC machine. Every NC machine definition must include one or more of these components.
- **Guide**—Wire guides on a wire EDM machine.
- **Other**—Any non-moving component not represented by one of the preceding types. This component type is often used to define housings, gear boxes, shields, etc.

Component Tree—Tree of defined components and models. Indentation indicates that a component is connected to a parent component. Component names are listed in bold text with connect position values. Models added to a component are listed beneath the component name. Use the + / - symbols to expand or collapse branches of items in the tree.

Tips:

Features used to manipulate components are displayed in a shortcut menu when you right-click on a component in the Component Tree window. Similarly, features used to manipulate models are displayed when you right-click on a model in the tree.

Click repeatedly on a model in the graphics area to toggle selection between the model and its component. Refer to the Component Tree window to review what is selected.

Using the center mouse button of a three button mouse, click repeatedly on a model in the graphics area to toggle selection between different models along the line of sight from the pick point. Use this technique to select models hidden from view by an object in the foreground.

Delete Material

Location: **Model menu > Delete Material**

Toolbar shortcut: 

Opens a window to delete or keep pieces of stock material. (See how) This feature is typically used to delete pieces of excess stock that have been cut free from the workpiece. However, pieces of other displayed models can also be deleted.

Delete Material window features:

Delete—Indicates selected items are to be deleted.

Keep—Indicates the selected item is to be kept. VERICUT will delete model pieces that are not connected to the selected piece of the model.

Items selected for deletion are erased from the active view immediately. However, erased material is not actually deleted from all views until you click **OK**. Clicking **Cancel** cancels the deletion and restores items that were erased for deletion.

Note: Once deleted, model pieces can not be restored.

Setup menu functions

Toolpath (Setup)

Location: **Setup menu > Toolpath**

Toolbar short cut: 

Opens a window to specify the tool path file(s) to simulate, and controls how VERICUT receives descriptions of cutting tools. In general, VERICUT can obtain descriptions of cutting tools by processing parametric cutter description records found in the tool path file, or can be configured to retrieve them from a VERICUT Tool Library file.

Toolpath window features:

Toolpath Type—Type of Tool Path file for VERICUT to simulate. By default, VERICUT simulates G-code or "machine code data" tool paths destined for a 3-axis mill machine with a "Fanuc-like" control. Using this option, you can quickly configure VERICUT to simulate tool paths from all popular CAM systems. Options follow.

- **G-Code Data**—G-code tool path file. With this choice, VERICUT uses a Machine file, Control file, and G-Code settings (ref. **Setup menu > G-Code > Settings**) to interpret codes present in the file.
- **VERICUT APT**—Generic ASCII APT tool path file from virtually any source, including ACL format.
- **UG CLS**—Unigraphics CLS tool path file.
- **CATIA APT**—CATIA APT source tool path file. This file must be processed from CATIA using the CATUTIL function and typically has a ".APTSOURCE" extension.
- **CV APT**—ComputerVision APT tool path file.
- **CADRA APT**—CADRA APT tool path file.
- **APT (RevPost)**—ASCII APT tool path file created from reverse post-processing G-code data-ref. "Reverse post-process G-codes to APT" in the *Using VERICUT* chapter.
- **Pro/MFG APT**—Pro/Manufacturing APT tool path file.
- **NCL APT**—NCL APT tool path file.

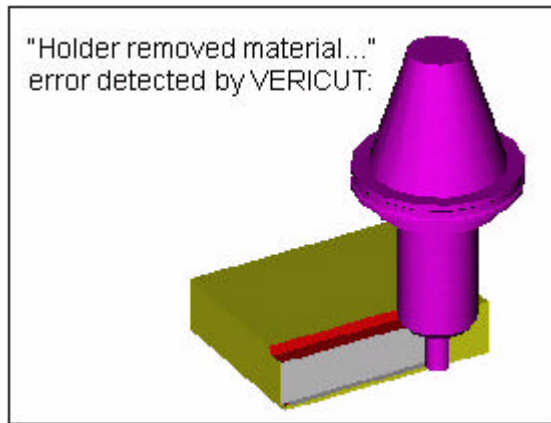
In addition to the tool path types above, VERICUT's converters enable you to simulate tool paths from various CAM systems. See the following sections for more information:

- File menu > Convert > Binary CL (*Reference* chapter)
- File menu > Convert > Mastercam (*Reference* chapter)
- Reverse post-process G-codes to APT (*Using VERICUT* chapter)

Tool Change By—Controls how VERICUT receives descriptions of cutting tools. Options are:

- **Cutter Desc.**—Process parametric cutter descriptions in the tool path file. Examples: CUTTER, TLDATA, and PPRINT/VERICUT-TC.
- **Tool Number**—Process tool path records that reference tool or pocket numbers and retrieve associated tools from a Tool Library file. Examples: TnM6 (G-code tool paths), LOADTL or LOAD TOOL, and TURRET.
- **List**—Refer to a list of tool change event-to-tool ID references that retrieve associated tools from a Tool Library file. The list is accessed via the **Use Tool List** feature, as described later in this section.
- **File Name**—Use a tool from a Tool Library file as specified for each tool path file in the list.

Use Holders—When active, instructs VERICUT to display defined tool holders and detect collisions they may have. Holders are non-modal and are applied only to the active cutter. If a holder removes material VERICUT issues an error. Material removed by the holder is shaded using the red **Error** color.



Initial Tool—When active, VERICUT loads an initial tool from a Tool Library file. This action occurs automatically when the User file is loaded, the tool path is rewound, or the model is reset. Choose the ID of desired tool using the option list next to this feature.

This feature is useful when simulating tool paths for machines that have a tool loaded at the time tool processing is started, therefore do not have information about the first tool.

Tool Override—When active, overrides the current cutter description with a specified tool from a Tool Library file. This action occurs immediately. Choose the ID of desired tool using the option list next to this feature. VERICUT uses the override tool until you select a different tool ID or clear the **Tool Override** checkbox. When cleared, the tool that would have been in use becomes the active tool.

Toolpath/Coord. System list— Also known as the "tool path list", lists tool paths to simulate with corresponding coordinate systems. In the case of changing tools via **File Name** (see above), this list also specifies the ID of a tool from the current Tool Library file to use with each tool path file.

Tool path file names can be entered directly, or managed using the buttons located below the list. Tool paths are processed by VERICUT in listed order. List order is changed via left-dragging the button at the start of the entry in the list. Only tool paths that have not been simulated by VERICUT can be changed. Reset the model to change tool paths that have been simulated.

Tip: Right-click in the list area to expose a shortcut menu for adding and deleting list items.

Add—Opens a window to add selected tool path files to the list. To select multiple tool path files for addition, use any combination of the techniques below:

- *Select multiple files in sequence*—Click first file name in the sequence, then press and hold the <Shift> key while clicking the last file name—all files in between those selected are also selected.
- *Select additional files*—Press and hold the <Control> key while selecting each additional file. Selecting twice unselects the file.

Replace—Replaces the selected tool path file in the list.

Delete—Deletes the selected tool path file from the list.

Clear—Clears the list of all tool path files.

Use Tool List—When **Tool Change By=List**, this feature opens a window to create and maintain a list of references that link tool change events to tools stored in a Tool Library file. You can use this list to supply tools from the library for sequential tool changes, or to replace tools specified for use in the tool path file. (See window features)

Tool Change List window features

Opened via pressing **Use Tool List** on the Toolpath window (see above), the features on this window maintain a list of references that link tool change events to tools stored in a Tool Library file. You can use this list to specify tools to use with sequential tool changes, or to replace tools specified for use in the tool path file.

Tool List window features:

List Tool Change By—Specifies the tool change event used to build the tool list from. Select an option, then press **Build Tool List** to have VERICUT automatically create the tool change event list. Options are:

- **Cutter Desc.**—Changes tools when parametric cutter description records in the tool path file are encountered, for example: CUTTER, TLDATA, and PPRINT/VERICUT-TC.
- **Pocket Num.**—Changes tools when tool path records that reference tool or pocket numbers are encountered, for example: TnM6 (G-code tool paths), LOADTL or LOAD TOOL, and TURRET.
- **Prog. Stop**—Changes when “program stop” codes are encountered, for example: M0 (G-code tool paths) and STOP.

Tool Change Event/Tool ID list—List of tool change events expected to occur during the simulation as indicated by the active **List Tool Change By** option (see above), and IDs of the tools to use for each. Event text must match the tool change record text exactly as it appears in the tool path file. Type or left-click and choose the tool ID from an option list. Right-click to add and delete tool list entries.

For the **Cutter Desc.** and **Prog. Stop** options, the list represents the sequential order of tool changes found in the tool path file(s), and tools to use for each. A **Pocket Num.** list is a cross-reference index between tool or pocket numbers referenced in the tool path file(s) to tool IDs in a VERICUT Tool Library.

Add—Adds a new tool change event to the tool change event list. Event data can be entered, or in the case of tool component IDs and OptiPath settings—can be selected from option lists.

Delete—Deletes the selected tool change event from the tool change event list.

Clear—Clears all tool change events from the list.

Prompt for OptiPath Settings while building the tool list—When active, causes VERICUT to prompt for OptiPath settings during interactive tool path optimization. At each tool change event an Optimization Settings window opens. Configure this window as desired to optimize the cutting sequence, or choose "**Do not optimize this tool**". An OptiPath Library file is created as you go containing the optimization settings defined for each optimized tool.

For more information about optimizing tool paths, see "About optimizing tool paths" in the *Using VERICUT* chapter.

Skip Duplicate Cutter Descriptions—When active, causes VERICUT to skip duplicate cutter descriptions when building the tool change event list. This feature is useful for suppressing redundant tool changes often output by CAM systems for different cutting operations by the same cutter.

Build Tool List—Builds a tool list by scanning the files in the tool path list and looking for the tool change events specified by the **List Tool Change By** feature (see above). When multiple tool path files are listed, a comprehensive tool change events list is created with all tool change activity from all listed tool paths.

Motion (Setup)

Location: **Setup menu > Motion**

Toolbar shortcut: 

Opens a window used to control the simulation of tool path and machine motions.

Motion setup window features:

Machining Type—Type of machining to be simulated. Options are: **Mill** (includes drilling and wore EDM) or **Turn** (includes drilling on spindle axis).

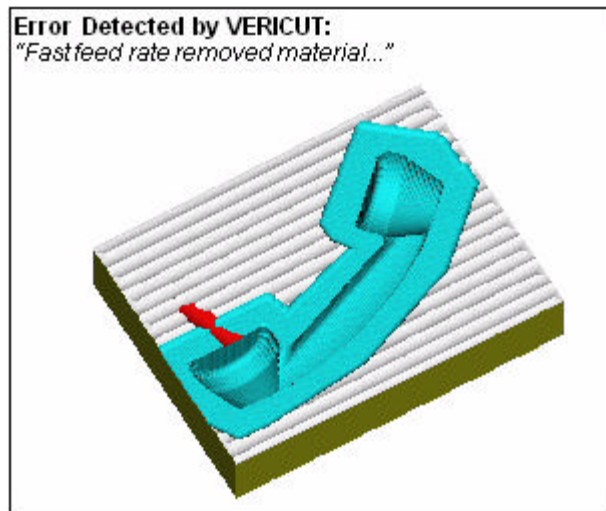
Motion:

Start At Line—Specifies the line, or tool path record at which cutting simulation starts. VERICUT fast forwards from its current location to the specified line while internally processing intermediate tool path records to store values for the current feed rate, spindle speed, coolant, tool description, and tool location.

Stop At—Provides options for stopping tool path processing. When applicable, enter supporting "stop at" text or values in the field next to this feature. Text entered is not case sensitive.

Stop At Max Errors / Stop At Max Warnings—Quantity of errors (or warnings) which, if detected, stops tool path processing.

Fast Feed—Feed rate threshold at which material removal is unsafe (likely to cause tool breakage, machine malfunction, or part damage). Material removed under these conditions is shaded in the **Error** color (typically red) and causes a "Fast feed rate" error.



Skip Cut—Quantity of animated tool motions to skip before updating the display. Enter "0" to display each tool motion. Skipping cuts increases animation speed since the display is not constantly updated. The cut model is unaffected.

Drill Cycle—Controls when and how to simulate canned tool axis cycles (e.g. G8n, "CYCLE"). Options are:

Help on VERICUT

- **Ignore**—Ignores cycle motions- the tool moves to each cycle position without performing cycle motions.
- **Full Motion**—Simulates all cycle motions, including pecking or chip-breaking motions.
- **Bottom Only**—Simulates cycle motions for a hole with one full depth motion- pecking or chip-breaking motions are not simulated.

While VERICUT simulates most popular cycle motions, there are also capabilities to add more and change how existing cycles are interpreted- see "About configuring for cycle simulation" in the *Using VERICUT* chapter for more information.

Need help troubleshooting cycle motion errors? See the *ERRORS and Problem Solving* chapter for assistance with diagnosing error messages and conditions.

Animation Speed:

Animation Speed slider—Controls the speed at which VERICUT animates material removal in workpiece views. Moving the sidebar left slows animation by adding intermediate tool display positions between motion start and end points. Moving the sidebar right reduces the number of intermediate tool display positions to speed animation display.

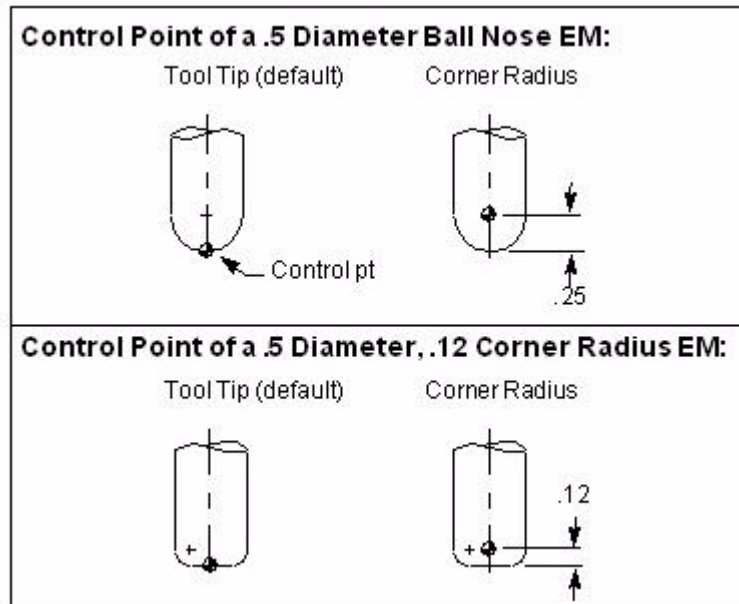
Tip: To group cuts together for even faster animation display, use the **Skip Cut** feature (see above).

Control Machine Simulation Speed—When active, applies **Animation Speed** control to machine movements in machine views. Clear this checkbox for machines to move as fast as possible.

Tool:

Tool Display—Controls when and how the tool is displayed. Options are: **On** (shaded solid), **Translucent**, or **Off** (do not display the tool). Material removal is not affected by this choice.

Control Point—Controls how milling tools are driven. By default, VERICUT drives the tool tip. Material is removed based on tool path motion commands and the tool shape, relative to the control point. Options are: **Tool Tip**, or **Corner Radius** (also known as "ball center").



Notes:

- If **Corner Radius** is used with non-ballnosed endmills, the offset is calculated along the tool axis, from the tool tip to the center of the first arc encountered in the tool profile.
- To specify a milling tool control point other the tool tip of corner radius, define tools via the Tool Manager and specify a control point offset via the tool's **Properties**.
- This feature is not applicable to turning tools. Instead, define tools via the Tool Manager and specify a control point offset via the tool's **Properties**.

Cut Color Method—Method of determining colors used to shade machined cuts. Options are: **Next Available** (determined via the **Cut Seq. / Color** list on **File menu > Colors: Assign** tab), or **Tool Library** (determined by the **Color** stored with tool properties in a VERICUT Tool Library).

Min. Cutter Height—Default height used when the current tool description does not specify height, or when specified height is less than this value. Affects milling cutters only.

Calculate Min. Cutter Extension—When active, calculates the minimum height required to avoid shank/holder collisions. This feature is effective only in FastMill cut mode with tool holders in use.

Minimum cutter extension calculations begin with the defined cutter height. If potential holder collisions are detected during the simulation, VERICUT extends the cutter to avoid the collision and adds this amount to the calculated height. Note that when these calculations are being performed, tool holder collisions are not reported since VERICUT extends the cutter to avoid the collision.

If desired, you can include additional clearance in the calculation via specifying **Holder Clearance** (see below). The final calculated height for each tool is displayed in the Log file Tool Summary section under the "Min" column heading.

Holder Clearance—Specifies clearance to add when calculating minimum cutter extension (see above). The clearance value is also included in the Log file minimum height values.

Settings (G-Code)

Location: **Setup menu > G-Code > Settings**

Toolbar short cut: 

Opens a window to configure settings to support processing G-code tool path files. This window looks different, depending on the tab that is selected. See the appropriate section below for specific information.

G-Code Settings window features:

Settings tab features

Job Tables tab features

Tool Tables tab features

Variables tab features

Subroutines tab features

Block Skip tab features

Turning tab features

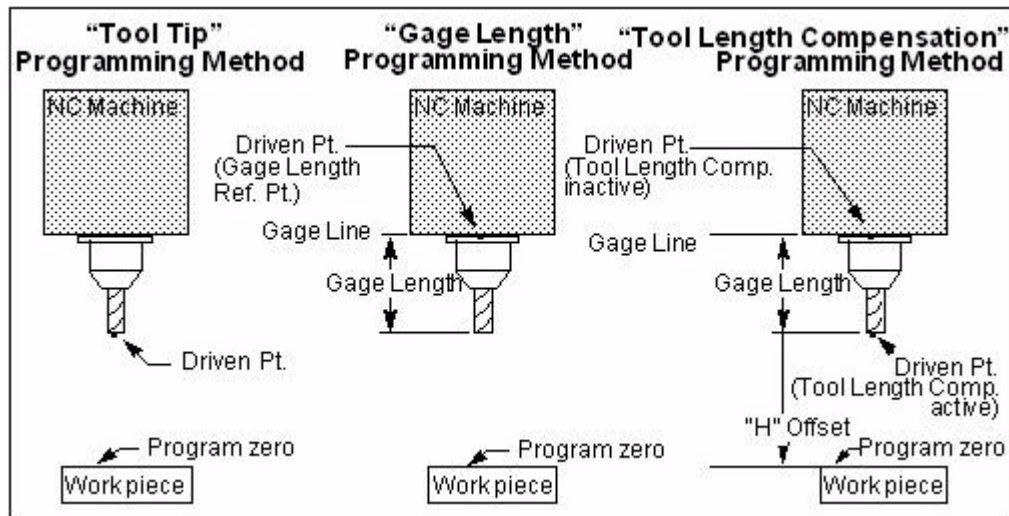
WireEDM tab features

G-Code Settings "Settings" tab features

Programming Method—Describes the relationship between cutting tools and what is being driven by tool path file motion commands. (Take care NOT to describe the relationship between cutting tools and the driven point controlled by a *CAM system*.)

Options are:

- **Tool Tip**—Drives the center of the tool tip.
- **Gage Length**—Drives the gage length reference point, where the centerline of the tool intersects the machine gage line. Gage length values must be supplied, as described by tool properties or via entries in a Gage Offset table.
- **Tool Length Compensation**—Drives the tool tip when length compensation is active and drives the gage length reference when length compensation is cancelled. Supply gage length values as described above for **Gage Length** programming method. Tool length compensation values are automatically calculated when an Input Program Zero table is used. However, you can manually enter, or override automatically calculated values via using a Tool Length Compensation table. (Ref. **Setup menu > G-Code > Settings: "Tool Tables" tab**)



Process Cutter Comp.—When active, applies cutter compensation, or "CDC" offsets to the programmed tool path. Enter CDC offset values in a Cutter compensation table, as described under "Cutter compensation table" in the *Reference* chapter. Clear this checkbox to ignore CDC-related codes in the tool path file.

Notes:

- **Output Precision** specifies the accuracy for calculating CDC offset tool positions. (Ref. **Setup menu > Control > Settings > Motion**).
- Tool paths can not be optimized by OptiPath when CDC offsets are being simulated.

Scan Toolpath Files—When active, performs a "scan pass" on tool path files to identify jump destinations for looping-branching logic. Tool paths with this type of logic must be scanned for proper simulation in VERICUT.

Ignore Data after Column—Specifies the G-code data column number to begin ignoring remaining data in the block. Use this feature when unmarked comments are permitted in designated columns of the G-code data. Enter the number of the last valid G-code data column in the field next to this feature.

Rotary Feed Motion Tol—Tolerance used to calculate intermediate tool positions resulting from processing rotary motion when **Output Intermediate Points** is active (ref. **Setup menu > Control > Settings > Rotary**). Decreasing this value increases the quantity of intermediate points calculated, and decreases simulation speed.

Rotary Rapid Motion Tol—Similar to **Rotary Feed Motion Tol**, except applies to rapid rotary motions.

G-Code Settings "Job Tables" tab features

Job Tables are used to specify work offsets and machine locations specific to the current tool path file(s). For additional information about tables, see one of the following topics:

What do you want to do?

Help on VERICUT

Learn about configuring tables

See how to add, modify, or delete tables

See reference information about tables, such as: list of tables, how they are used, data entry, etc.

G-Code Settings "Tool Tables" tab features

Tool Tables are used to store tool-related offset and compensation data. For additional information about tables, see one of the following topics:

What do you want to do?

Learn about configuring tables

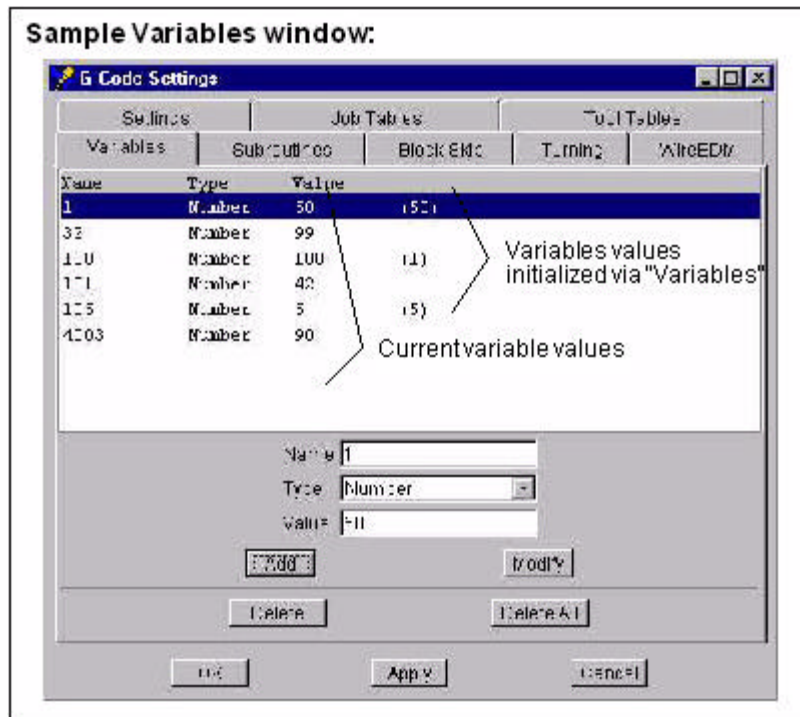
See how to add, modify, or delete tables

See reference information about tables, such as: list of tables, how they are used, data entry, etc.

G-Code Settings "Variables" tab features

Features on this tab are used to initialize, maintain, and monitor variables that are referenced by the tool path file. Variables are assumed to have zero for a default value. However, you can initialize variables with any numeric or text value. (See how)

Variables list—List of variables initialized and encountered during tool path processing. Variables initialized via this function are shown in parenthesis. (See figure below)



Name—Name of the variable. Each name must be unique.

Type—Type of variable; also indicates the type of data that can be assigned to the variable. Options are: **Number** (real number), or **Text** (text string). Examples follow.

Examples- Assign of value of “0010” to a **Number** type variable, and a numeric value of “10” is retained. However, assign “0010” to a **Text** variable type, it retains the text characters “0010”.

Value—Variable value.

Add—Adds, or initializes a variable.

Modify—Modifies the selected variable to have the data configured in the Variables window.

Delete / Delete All—Deletes the selected variable (or all variables).

Tips:

- When the Variables window is left open during processing, it gives immediate feedback when variable values are set or changed.
- Debug messages can be used to document when variables are set during processing- ref. **Setup menu > G-Code > Process Options > Debug**.

G-Code Settings "Subroutines" tab features

The features on this tab are used to specify names of external files containing subroutines that can be accessed by VERICUT. For additional information about subroutines, see one of the following topics:

What do you want to do?

Learn about simulating subroutines

See how to add, modify, or delete subroutine file names

Configure "Subroutines" control settings

G-Code Settings "Block Skip" tab features

Features on this tab control which levels of block skipping are recognized, and specify the block skip character. (See how)

Block Skip Character—Character which causes the block to be skipped. This character must be the first character in the data block for skipping to occur.

Apply Switch Value—Controls when block skip switch values are referenced by VERICUT. Options are:

- **Immediately**—Reference switch settings when VERICUT is reset, upon loading a new User file, and any time tool path processing is started.
- **At Reset**—Reference switch settings only when VERICUT is reset, or upon loading a new User file.

Switch 1-9—When active, processing is skipped for blocks beginning with the designated block skip switch. For example, "Switch 2" corresponds to blocks beginning with "2/", "Switch 3" corresponds to blocks with "3/", and so on.

G-Code Settings "Turning" tab features

Features on this window are used to configure settings applicable to simulating turning tool paths.

4-Axis Sync Method—Controls when and how to interpret turret/head synchronization data. When active, 4-axis turning motion synchronization codes is processed so that motion components travel with the same timing as occurs on the actual NC machine tool. The active choice indicates the NC control style of 4-axis synchronization, also known as "merged turning".

Options are:

- **None**—(Default) processes blocks as they are encountered-primary axes are not synchronized with secondary axes. Only one tool cuts at a time.
- **G&L**—"N" sequence numbers mark commands for the primary axes (typically XZ), while "O" sequence numbers mark commands for secondary axes (typically UW). "B" sets the sync code, and "M12" synchronizes primary axes with secondary axes.
- **Okuma**—"G13" mode commands move the primary axes, while "G14" mode commands move the secondary axes. "P" sets the sync code.
- **Fanuc**—Two independent tool path files are simulated. Tool path file #1 move the primary axes, while Tool path file #2 moves the secondary axes. "Mnnn" sets the sync code, and "G68" synchronizes primary axes with secondary axes.

When synchronized tool path motions are processed, the **Info menu > Toolpath** window shows

which blocks are being processed by the primary and secondary axes.

Primary SubSystem ID / Secondary SubSystem ID— 4-axis turning machines intended to simulate synchronized motions must be configured with two subsystems: one for the primary axes, a second for the secondary axes. These features associate the G-codes (or tool path files) which command primary motion axes with the primary machine subsystem, and the secondary motion axes/subsystem .

Note that feature labels dynamically change, based on the active **4-axis Sync Method**.

Active 4-axis Sync Method:	Primary/Secondary SubSystem ID features:
None	Primary SubSystem ID, Secondary SubSystem ID
G&L	N Block SubSystem ID, O Block SubSystem ID
Okuma	G13 SubSystem ID, G14 SubSystem ID
Fanuc	File 1 SubSystem ID, File 2 SubSystem ID

CSS Maximum RPM—When constant surface speed, or "CSS" is controlling spindle speed, this value specifies the fastest RPM to turn the workpiece.

RPM Changes with RAPID and CSS—When active and CSS is controlling spindle speed, rapid motions (e.g. G0) can affect spindle speed, as well as linear and circular motions (e.g. G1-3). Clear this checkbox to have only linear and circular motions affect RPM.

G-Code Settings "WireEDM" tab features

Features on this tab provide settings that must be configured to simulate wire EDM machining. (See how)

Wire Diameter—Specifies the effective cutting diameter of the wire.

Work Table to XY Plane—Distance from the top of the work table (machine Z zero) to the XY reference plane, as measured along the machine Z-axis. This value is usually zero, but can be a positive or negative value. The XY reference plane is the Z-axis height at which the tool path file X and Y values are driven.

XY Plane to UV Plane—Distance from the XY reference plane to the UV reference plane, as measured along the machine Z-axis. This value must be a positive value. This setting is only used when the tool path file contains U and V-axis values to control the wire angle. The UV reference plane is the Z-axis height at which the tool path file U and V values are driven.

Output Files

Location: **Setup menu > G-Code > Process Options > Output Files**

Opens a window to control conversion, or "reverse post-processing" of G-code tool paths, and specify the names of files resulting from this process. (See how)

G-Code Output Files window features:

APT Output File—Name of the file to receive ASCII APT tool path records resulting when **Create APT Output File** is active (see below). If blank, the file is not created.

G-Code Log File—Name of the G-Code Log file to receive error, warning and informational messages about G-code processing. If blank, the file is not created.

Create APT Output File—When active, converts the G-code tool path data into ASCII APT tool path records. Converted data is stored in the specified APT Output file (see above).

Messages

Location: **Setup menu > G-Code > Process Options > Messages**

Opens a window to set the destination and format of messages output during a simulation. (See how)

Messages can be sent to the destinations listed below, however, not all choices are available for all message types.

- to Log
- to Screen
- to Log and Screen
- to Output
- to Output and Log
- to Output and Screen
- to Output, Log and Screen
- to Nowhere

Messages window features:

Send Error Messages—Destination of messages about errors-potentially severe problems that will most likely produce incorrect results on the machine tool.

Send Warning Messages—Destination of messages about possible problems. Scrutinize warnings to decide if they will actually impact the machining process.

Send Info Messages—Destination of messages providing reference information.

Send Debug Messages—Destination of debug messages providing additional information related to internal calculations, conditions, and variable values. The types of debug messages which appear are controlled via **Setup menu > G-Code > Process Options > Debug**.

Send Modal Messages—Destination of messages showing the control settings that were in effect when the tool path file was processed.

Send Input Messages—Destination of messages showing the input tool path file records processed.

Send Output Messages—Destination of messages showing the converted tool path records resulting from reverse-postprocessing. When **Create APT Output File** is active in the **Setup menu > G-Code > Process Options > APT Output** function, output messages are sent to the APT Output File specified on the same window. By default, these records are in ASCII APT format. However, using the Machine Developer's Kit, other output file formats are obtainable.

Output Filenames Format—Controls how names of files used by VERICUT are shown. Options are: **Filename Only** or **Full Path Names** (includes full directory path).

Output Block Sequence Numbers—Controls when block sequence numbers are included with APT output records resulting from conversion. The sequence numbers appear in columns 73-80,

Help on VERICUT

and are useful for general reference or debugging purposes.

Output "Word is not defined" Errors—When active, outputs the “Word is not defined” error message when G-code words are encountered that the control is not set up to process. To prevent these errors from appearing, ensure this feature is not active.

Debug

Location: **Setup menu > G-Code > Process Options > Debug**

Opens a window to set debug messages to output, and where to send them. (See how) Debug messages can provide additional information that may be helpful debugging development of files used to simulate G-code tool paths and simulate machine tool motions.

Note: The output destination for debug messages is controlled by "**Send Debug Messages**" on the **Setup menu > G-Code > Process Options > Messages** window.

Debug window features:

Debug Offsets—When active, outputs debug messages about offset value changes, including: coordinate system offsets/shifts, compensation values in use, etc.

Debug Delta Information—When active, outputs debug messages about rotating the tool position.

Debug Rate Calculations—When active, outputs debug messages about rate calculations related to tool movement.

Debug Loop Logic—When active, outputs debug messages about values calculated by looping or branching logic.

Debug Action Queue—When active, outputs debug messages about values stored in the CDC action queue.

Debug Intersect Points—When active, outputs debug messages related to intersect points calculated by CDC calculations.

Debug Cutter Compensation—When active, outputs debug messages about CDC tool offset positions.

Debug Cycle Values—When active, outputs debug messages related to processing canned tool axis cycles.

Debug Substitution—When active, outputs debug messages related to string substitutions (ref. **Setup menu > Control > Adv. Options: Substitute** tab).

Debug Variables—When active, outputs debug messages about changes to variable values. See also: **Setup menu > G-Code > Settings: Variables** tab.

Debug Turning—When active, outputs debug messages associated with turning, for example the distance from the turning axis to the tool control point and calculations related to the use of constant surface speed.

Debug Wire EDM—When active, outputs debug messages associated with wire EDM machining.

User Defined (G-code process options)

Location: **Setup menu > G-Code > Process Options > User Defined**

Opens a window to configure user defined control settings supplied via a custom CME file. This window will appear differently, depending on the CME file in use.

CME files are created using the Machine Developer's Kit, and specified for use in VERICUT via **Dev Kit CME** in the **Setup menu > Control > Adv. Options** function.

Note: Developer's Kit-defined settings can also appear under **Setup menu > Control > Settings > User Defined**, as determined by the developer.

View APT Output File

Location: **Setup menu > G-Code > Process Options > View APT Output File**

Opens a window containing the APT Output file with ASCII APT tool path records created when **Create APT Output File** is active (ref. **Setup menu > G-Code > Process Options > Output Files**).

For information about APT tool path records and formats, see "Supported APT-CLS records" in the *VERICUT Files* chapter.

View G-Code Log File

Location: **Setup menu > G-Code > Process Options > View G-Code Log File**

Opens a window containing the G-Code Log file. This file contains information, such as error, warning and informational messages about G-code processing. In this window, you can view, edit, or reset the G-Code Log file contents.

APT Settings

Location: **Setup menu > APT Settings**

Opens a window to configure settings to support processing APT-CLS tool path files. This window looks different, depending on the tab that is selected. See the appropriate section below for specific information.

APT Settings window features:

Motion tab features

Rotary tab features

Turning tab features

Cycles tab features

APT Settings "Motion" tab features

Default Toolpath Units—Default unit measurement system for tool path files that do not contain a "UNITS" record. If the tool path file units differ from the session units, values in the tool path are converted appropriately while being processed.

MULTAX—When active, interprets 6 parameter "GOTO" records as GOTO/x,y,z,i,j,k where i,j,k is tool axis orientation (superseded by MULTAX/OFF). Clearing this checkbox interprets these records as GOTO/x1,y1,z1,x2,y2,z2 -two point locations (superseded by MULTAX/ON).

Max Tool Axis Angle—Maximum angle (in degrees) along a multi-axis motion to display intermediate tool positions. A value of "0" (default) does not display intermediate positions on multi-axis motions. The cut model is unaffected by this feature.

Tool Axis Reset Angle—Minimum angle (in degrees) which is considered a positioning move. Multi-axis motions having angles equal to or greater than the reset angle do not remove material. Instead, the tool display is turned off through the multi-axis motion, then turned on after the multi-axis destination is reached. Cutting resumes at the new tool position.

Process Circles—When active, processes "CIRCLE" records to simulate the arc motion. Clear this checkbox to ignore "CIRCLE" records, such as necessary when CIRCLE records are accompanied by their representative chordal GOTO points.

Note: Simulating circle motions increases performance dramatically over processing chordal GOTO points.

Reverse Circles—When active, reverses circle motions. Clear this checkbox to simulate circle motions in the direction programmed. (Circle IJK values determine circle direction.)

Detect Chordal GOTOs—When **Process Circles** and this feature are active, the chordal GOTO points which follow CIRCLE motion records are processed to simulate an arc, or circle motion. Clear this checkbox to process the GOTOs as linear tool positions.

Reset Tool Change—When active, resets the tool display following a tool change. This action turns off the tool display until after the tool is changed and the following motion is processed. The

new tool appears at the end of the motion following the tool change. This option is recommended when simulating tool paths for NC machines that automatically retract the tool during a tool change.

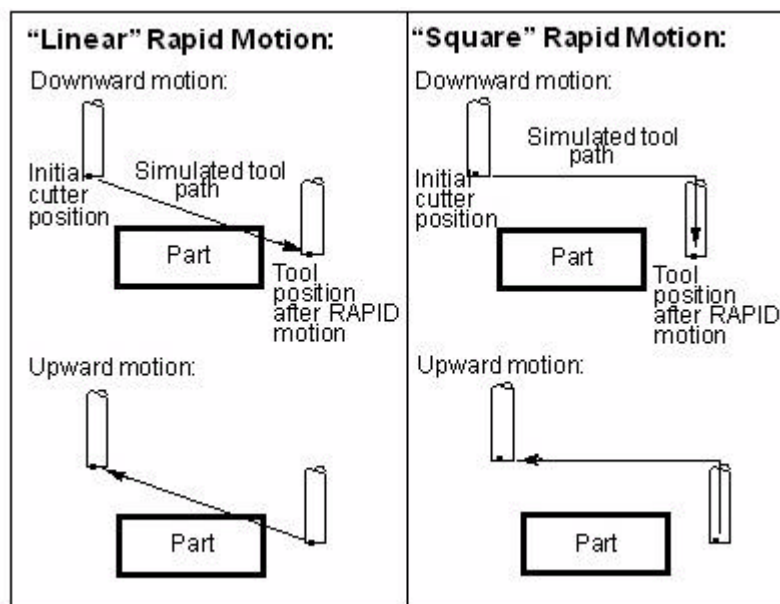
Clear this checkbox to have VERICUT change the tool at its current location when a tool change command is encountered.

Process Matrix—When active, processes these tool path matrix records to transform the tool path to its base coordinate system: CATIA0, MSYS, or PPRINT/VERICUT-MATRIX. Clear this checkbox to ignore tool path matrix records.

Hint: This feature is useful to automatically orient tool paths to models imported from the same CAD/CAM system.

Default CLEARP—Distance to retract along the tool axis when a “RETRACT” record is encountered prior to establishing the retract plane. The default CLEARP condition is “off” until a “CLEARP” record defining a clearance plane is encountered.

RAPID Motion—Controls how to simulate rapid positioning motions. Options are: **Linear** (move in a straight line), or **Square** (move with “squared off” motion, where Z motions are processed independently from X and Y).



GOHOME Motion—Controls how the tool is returned to the home position when a “GOHOME” record is processed. Options are: **FROM** (move in a straight line to the home position), or **Retract** (move with squared off motion, where the tool is first retracted the distance specified by the **GOHOME R Dist** value, then moved to the home position).

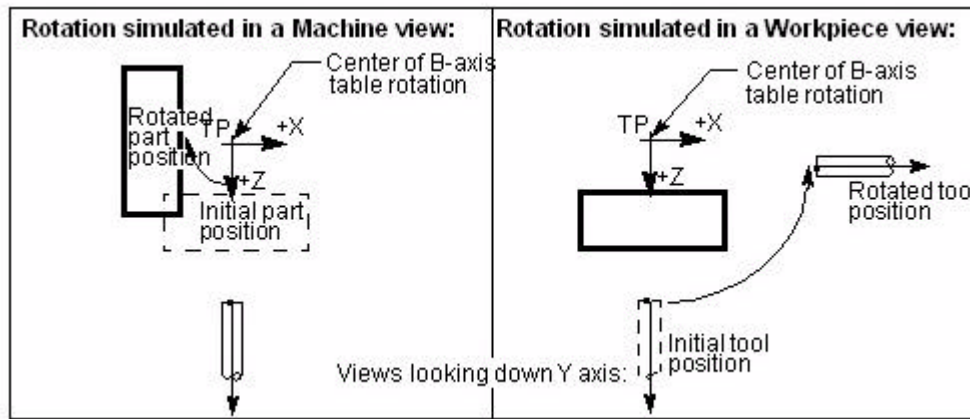
GOHOME R Dist—When simulating “GOHOME” motions via the “Retract” option (see above), this value specifies the distance to retract before going to the home position.

APT Settings "Rotary" tab features

VERICUT simulates rotation differently, depending on the type of view you are watching: machine vs.

workpiece. Machines typically move the workpiece, while in a Workpiece view the workpiece is fixed and the tool is seen rotating. Both simulated rotary cut paths are identical, and will remove material and detect collisions equally as well.

“ROTATE/BAXIS,ATANGL,90” simulated in a Machine view and Workpiece view:



ROTATE, ROTABL, ROTHED—These features control when and how to simulate motion caused by the corresponding rotary tool path record. “BAXIS” is the default rotation axis when not specified. The rotary pivot point is defined by the **Rotation Center**. When absolute or incremental rotation is not specified, the **Angle Value** determines how to apply the rotary value.

Options are:

- **Ignore**—Ignores the rotary command. (Default)
- **Rotate**—Rotates while cutting. Tool path motions that follow are expected to have the rotation applied.
- **Rotate/Apply**—Similar to **Rotate**, except also applies the rotated coordinate system to subsequent tool path motions. Tool path motions that follow are NOT expected to have the rotation applied.
- **Position**—Rotates without cutting, also known as "positioning". The tool disappears from its initial position, then appears at the rotated position. Tool path motions that follow are expected to have the rotation applied.
- **Position/Apply**—Similar to **Position**, except also applies the rotated coordinate system to subsequent tool path motions. Tool path motions that follow are NOT expected to have the rotation applied.

Rotation Center—Center of rotation for rotary commands processed via the **ROTATE**, **ROTABL**, or **ROTHED** features (see above).

Philosophy—Describes what is being commanded to rotate. Options are: **Table** (default), or **Tool**. Changing the rotation philosophy has the affect of reversing rotation direction.

Note: Rotation via “ROTHED” is always simulated using "Table" philosophy.

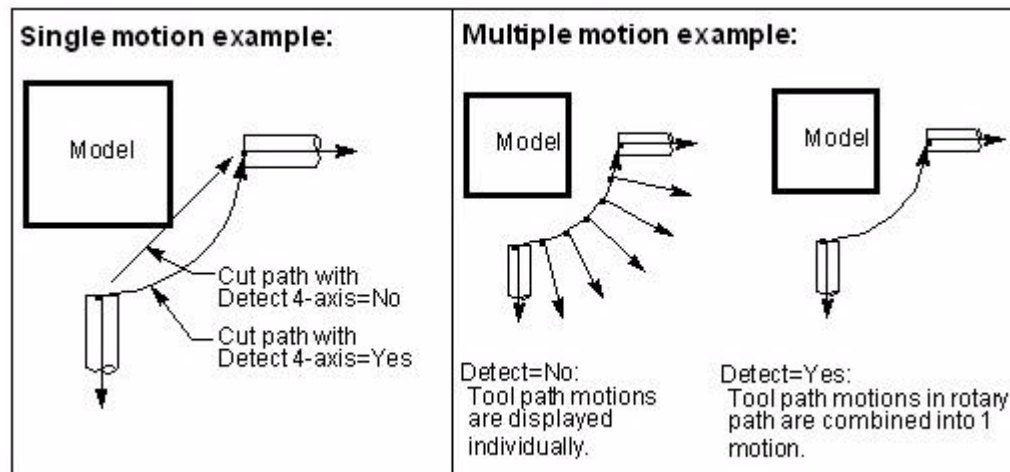
Angle Value—Controls how angle values are applied when simulating rotary commands that do not specify “ATANGL” or “INCR”. Options are: **Relative** (incremental angle values), or **Absolute** (default- absolute angle locations).

Note: "ROTHED" angle values are always assumed to be "**Absolute**".

Direction—Direction of rotation. Options are: **Shortest** (default- direction which travels the shortest distance), CW (always clockwise), or CCW (always counterclockwise).

Detect 4-Axis Rotary Motion—When active, simulates an arc cut for multi-axis motions that fit within an arc path. Clear this checkbox to simulate each multi-axis motion independently- recommended when simulating 5-axis contouring-type motions.

The **Interpolation Tolerance** value is used to determine which motions are simulated in the 4-axis cut (ref. **File menu > Properties: Tolerance** tab).



APT Settings "Turning" tab features

GOTO Format—Specifies how turning GOTO record values are to be displayed in a Workpiece view. In this view, VERICUT simulates turning motion in the ZX plane, where Z is the spindle-axis, and X is the cross-slide-axis. This feature acts like a post-processor for turning tool position data.

Example 1- Assume you have a turning tool path programmed in GOTO/x,y format. By default, **GOTO Format** "#zval, #xval" displays the first GOTO value as Z-axis motion, and the second value as X-axis motion.

Example 2- Assume you have a turning tool path that was reversed post-processed from G-code data such that GOTO records are in GOTO/x,0,z format (center value is "zero"). For VERICUT to process this correctly, set **GOTO Format**= #xval, #yval, #zval.

APT Settings "Cycles" tab features

Cycle Set—Controls which cycle definition set(s) are considered by VERICUT when processing APT tool paths. If you add a new cycle set, for a different flavor of APT or a different post-processor, you can give it any name you wish and it will appear in this list. "ALL" considers all cycle sets defined for use.

Help on VERICUT

Cycle Definitions—Opens a window to maintain cycle and modals definitions used to interpret “CYCLE” records. This feature is especially useful if simulated cycle motions are not displayed as expected. When VERICUT can not find definitions to interpret a CYCLE record, an error similar to “Invalid CYCLE record” is issued. In these cases, cycle definitions must be modified to interpret the cycle data.

For more information about configuring VERICUT to interpret cycle commands, see "About configuring for cycle simulation" in the *Using VERICUT* chapter.

Coord. System

Location: **Setup menu > Coord. System**

Toolbar shortcut: 

Opens a window to define a coordinate system. (See how) The active coordinate system applies to X-caliper measurements, Section plane values, and tool path motions (except when orientations are defined in a tool path list). To see axes representing the active coordinate system (XcsYcsZcs), select **View menu > Axes: Coordinate System**. Coordinate system axes only appear in the workpiece view.

A user-defined coordinate system is typically used to locate an APT or CLS tool path for proper relationship to the workpiece. You can also instantiate different coordinate systems for defining section planes or gathering measurement data.

Note: Coordinate systems are not recommended to locate G-code tool paths on a simulated 3-D machine, as this can cause the workpiece to move in the machine view. Instead, use an Input Program Zero table, or work offsets if present in the tool path file.

Coordinate System window features:

Coordinate System Name—Adds a new coordinate system. You can enter a name into the top position of the list, or VERICUT will supply a unique coordinate system name for you.

Delete—Deletes the selected coordinate system from the **Coordinate System Name** option list.

Orient:

Choose one of the following **Orient** methods, then click the arrow and follow the prompts in the message area to define a coordinate system via picking three points, X and Y axes, or three planes.

3 Points—Defines a coordinate system via picking three points: origin point, point on the X-axis, and point on the Y-axis.

X-Axis, Y-Axis—Defines a coordinate system via picking the X and Y axes.

3 Planes—Defines a coordinate system via picking three perpendicular locating planes.

Translate:

From / To / Move—These features translate the coordinate system origin. Translation is incremental, as calculated from the "**From**" point to the "**To**" point location. Incremental translation is applied each time **Move** is pressed.

XYZ values can be entered (separated by spaces), or filled in by clicking on a model. In general, graphical selection supports picking corner points and midpoints of model geometry. A crosshair and vector show you the pending pick-point location.

Tip: If the applied translation is incorrect, press **Undo** to return the coordinate system to its previous location.

Rotate:

Center of Rotation—Specifies XYZ center of rotation used when the rotation buttons are used

Help on VERICUT

(see below). XYZ values can be entered (separated by spaces), or filled in by clicking on a model, as described above for **From / To / Move**.



To see the center of rotation, press . To remove the center of rotation symbol press this button again, or close the Coordinate System window.

Increment—Specifies incremental degrees of rotation applied when one of the rotation buttons are pressed.

Rotation buttons—(X+/X-, Y+/Y-, Z+/Z-) Applies the incremental degrees of rotation specified in the **Increment** field. Rotation occurs about the **Center of Rotation** relative to the active component origin.

Tip: If the applied rotation is incorrect, press **Undo** to return the object to its previous location.

Location:

Position—Specifies the absolute XYZ position of the coordinate system relative to the machine coordinate system origin. Values in this field are separated by spaces.

Angles—Specifies the absolute XYZ rotation of the object relative to the machine coordinate system origin. Values in this field are separated by spaces.

Undo—Returns the coordinate system to its previous location, or as it was when the Coordinate System window was opened.

Note: Coordinate systems specified in the **Setup menu > Toolpath: Coord. System** list override the active user-defined coordinate system.

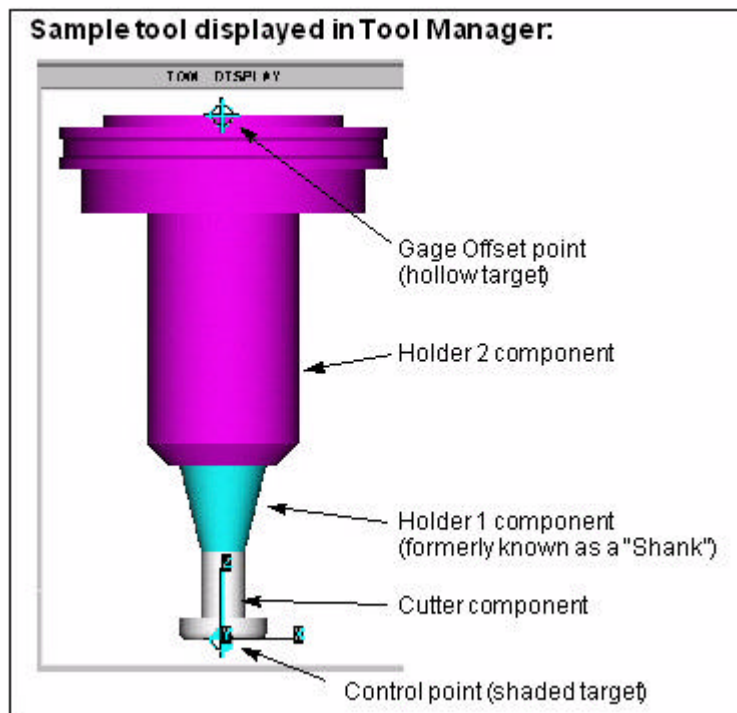
Tool Manager

Location: **Setup menu > Tool Manager**

Toolbar shortcut to access the Tool Manager: 

Opens a window to create and maintain Tool Library files containing descriptions of cutting tools, or tool assemblies. Tool Library data is used by VERICUT when the **Setup menu > Toolpath** function is configured to do so. Tools available in the library are listed by ID and description.

The Tool Manager shows the selected tool in the **Tool Display** area. Colors distinguish components in a tool assembly. Target symbols depict the tool control point and gage offset point. To learn more about describing tools for VERICUT, see "About describing cutting tools" in the *Using VERICUT* chapter.



Tool Manager window features:

File menu—Functions in this menu create a new Tool Library, open or save Tool Libraries, and exit the Tool Manager.

View menu—Functions in this menu provide control over viewing tools in the display area, such as: view orientation, lighting, and symbols used to depict control points, gage offsets, etc.

- **Orient**—Opens a windows orient the view of a tool, including: rotate, zoom, pan, reverse, etc. Features on this window are the same as those found on the **View menu > Orient** window.
- **Attributes**—Opens a window to control the attributes of the view of a tool, such as: symbol display (**Driven Point Offset**, **Gage Point Offset**, and **Axes**), light source, shading, etc.

Tool/ID list—List of tools available for use in the Tool Library file. Find tools you need by searching for tool ID or description text using the **Search** feature (see below).

Search, Forward/Backward—Searches for specified text in Tool Library record IDs and descriptions. Text entered is not case sensitive. Searching begins from a highlighted tool ID, and proceeds in the direction indicated by the button pressed (**Forward** or **Backward**). The first library entry found to match the search string is selected and the tool is seen in the display area.

Add—Opens a window to add new tools to the Tool Library. (See window features)

Modify—Opens a window to modify existing tools in the Tool Library. (See window features)

Copy—Opens a window pre-configured with values copied from the selected tool. If values are the same as required, all that remains is to add a unique tool ID. (See window features)

Delete—Deletes the selected tool from the Tool Library.

WARNING- If you delete a tool containing a referenced tool component, the deletion also occurs in tools which reference the deleted component.

Tool Add/Modify/Copy window features

Opened via pressing the **Add**, **Modify**, or **Copy** buttons on the Tool Manager window (see above), the features on this window describe a tool assembly, and how it will be used to machine the workpiece. (See how to add, modify, copy, or delete an existing tool)

ID—Identifies the tool in the Tool Library. Any number of characters can be entered, however, 15 are visible at a time in the **ID** field, and the first 23 can be seen in the Tool Manager tool list.

Description—Describes the tool or tool assembly. Any alpha-numeric text can be entered in this field. Special characters are not recommended.

Hint: Use IDs and descriptions that make it easy for others to find tools they need.

Tool Type—Type of tool being described. Available tool component shapes (icons) and properties are different, depending on the type of tool, or machining the tool will perform: **Mill** or **Turn**.

Units—Controls the unit measurement system (Inch or Millimeter) in which to describe the tool. If the tool units are different from the session units when loaded, then tool description values are converted appropriately.

Properties—Opens a window to define supporting tool properties, such as: gage length/offset, control point offset, color, etc. (See window features)

Tool Assembly area—Shows the components and associated shapes in the tool assembly. The **Tool Display** area shows a shaded image of the tool described by the data in this area.

Add, Modify, Delete—Adds/modifies/deletes tool components and shapes in the **Tool Assembly** area.

Position—Opens a window to orient 3-D turning tool shapes. (See window features)

Tool Component—Controls which component in the tool assembly is active for definition. Options are:

- **Cutter**—Cutting portion of the tool, often referred to as the "flute length" or "insert" portion of

the tool.

- **Holder 1**—Describes the non-cutting portion of the tool attached to the cutter. This portion of the tool was identified in pre-5.0 versions of VERICUT as a "shank".
- **Holder 2**—Describes the tool holder used to mount the tool to the NC machine tool. If all three tool component types exist, this component type connects to the **Holder 1** component. If a shank is not defined, then the holder is applied directly to the cutter.

Copy—Copies a tool component from a previously defined tool. You can choose from tools in the current Tool Library, or select “**External File**” to pick the desired component from tools in another Tool Library file. Only the tool data for the active **Tool Component** is copied.

Tool icons / Tool Description area—Use these icons to indicate the tool type you want to describe. Options are available to define tool shapes using parametric cutter descriptions, line-arc profiles, and copying or referring to tools in other assemblies. Selecting an icon configures the **Tool Description** area with the values needed to define the selected tool type.

After describing the tool component and shape, you must add it to the current tool description. Pressing **OK / Apply** at the bottom of the window does NOT automatically incorporate your changes in the current tool. Use the buttons in the **Tool Assembly** area to accomplish this task, after which you will see the updated tool assembly in the **Tool Display** area.

Tool icons and descriptions

Selecting an icon configures the **Tool Description** area with the values needed to define the selected tool type. Available icons differ, depending on the tool and component type being defined. The tables below list the available icons and corresponding descriptions, by tool type.

Mill tool icons:	Name:	Description:
	Flat Bottom End Mill	Diameter (D), Height (H)
	Ball Nose End Mill	Diameter (D), Height (H)
	Bull Nose End Mill	Diameter (D), Corner Radius (R), Height (H)
	Drill	Diameter (D), Drill Point Angle (A), Height (H)
	7 Parameter	D, R, E, F, A, B, H
	10 Parameter	D, R1, E1, F1, A, B, H, R2, E2, F2

	Profile	Point (X, Z), Arc (X Center, Z Center, Radius, Direction)
	Reference	Reference ID*
	None	Omit from tool assembly

* **WARNING** - If a referenced tool component is deleted, the deletion also occurs in all tool assemblies referenced to the deleted component.

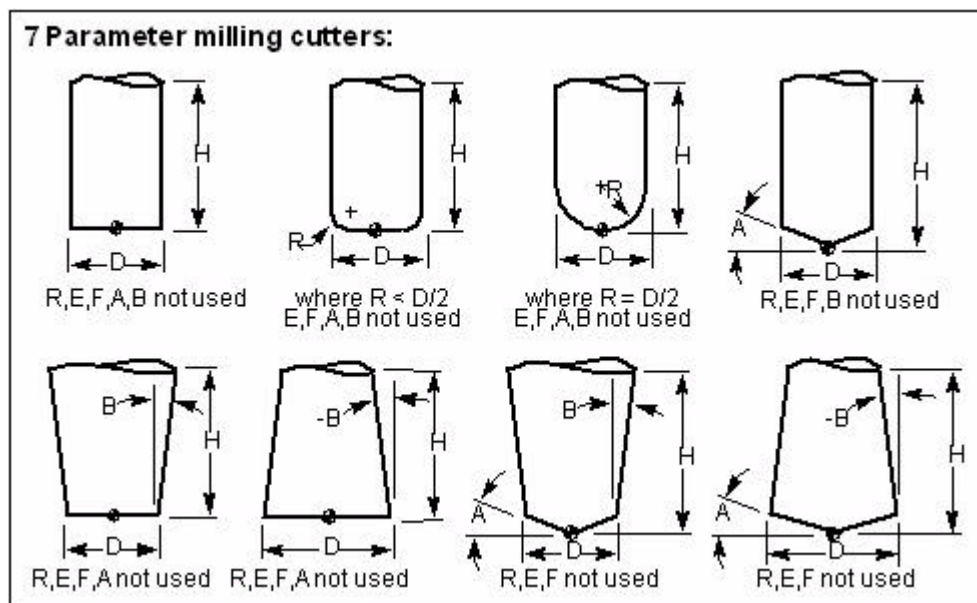
Tips for defining milling tools:

Diameter is measured from the intersection points of the cutter side with the bottom flat or angle.

D is automatically calculated when $D=0$, and the corner radius (R) and side angle (B) are specified. (Good tip for defining tapered endmills.)

E and F are automatically calculated when diameter and corner radius are specified.

Milling cutter shapes that can be defined via 7 parameters:



Turn tool icons:	Name:	Description:
	Triangle	Nose Radius (R), Inscribed Circle (IC), Lead Angle (A)

	Diamond	Nose Radius (R), Inscribed Circle (IC), Insert Angle (B), Lead Angle (A)
	Groove	Width (W), Radius (R), Length (L)
	Button	Diameter (D)
	Drill	Drill Point Angle (A), Diameter (D), Height (H)
	4 Parameter	Nose Diameter (D), Tool Angle (A), Tip 1/2 Angle (B), Height (H)
	Profile	Point (Spindle, Cross Slide), Arc (Spindle Center, Cross Slide Center, Radius, Direction)
	Block	Length (L), Width (W), Height (H)
	Cylinder	Radius (R), Height (H), Tolerance
	Cone	Bottom Radius (R1), Top Radius (R2), Height (H), Tolerance
	STL File	STL File, Normals
	VERICUT File	VERICUT File
	Reference	Reference ID*
	None	Omit from tool assembly

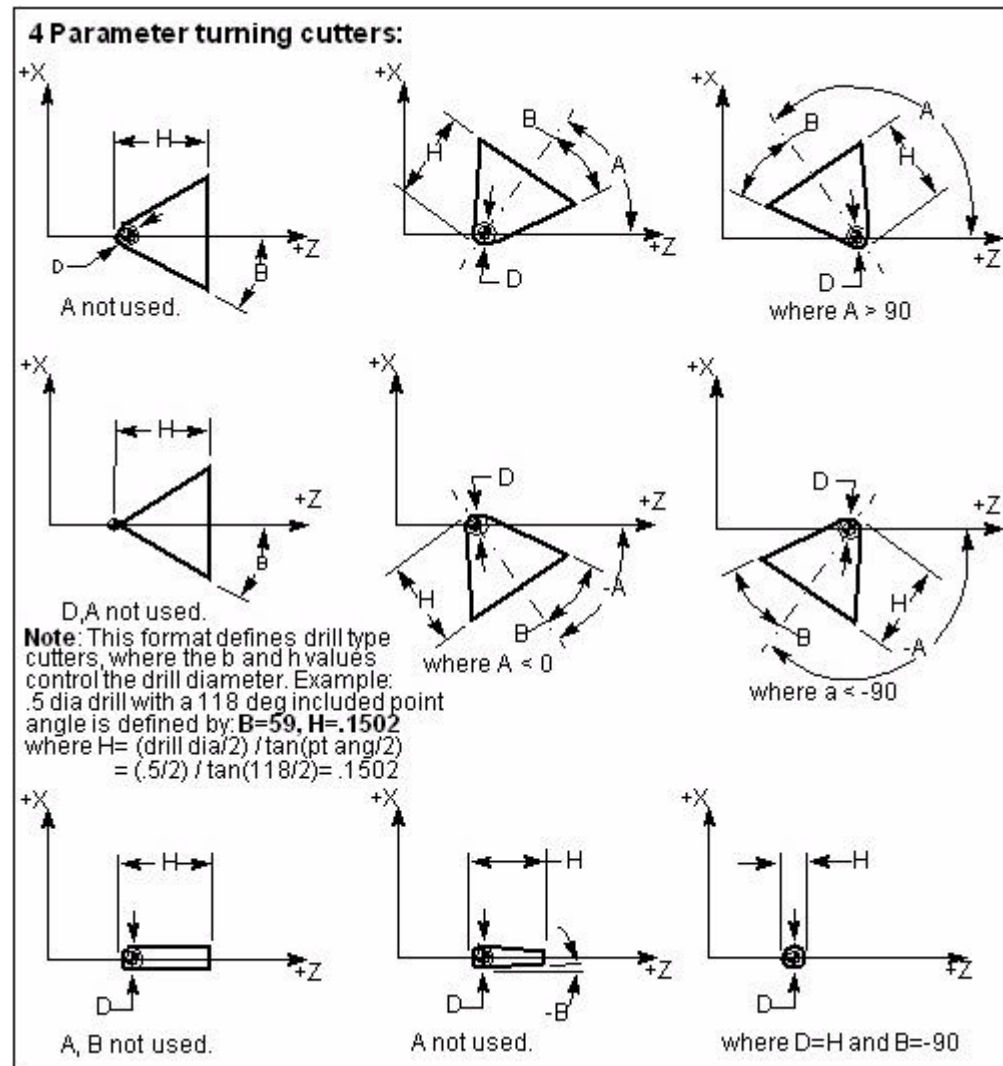
* **WARNING** - If a referenced tool component is deleted, the deletion also occurs in all tool assemblies referenced to the deleted component.

Notes for defining turning cutters:

Since all turning tool descriptions (except **Drill**) include a **Thickness** value to give the tool a 3-D appearance, this value is omitted from the turning tool descriptions in the table above. "0" causes VERICUT to internally calculate a thickness.

For turning drills, only the drill tip is displayed in a workpiece view.

Turning cutter shapes that can be defined via 4 parameters:



Tool Manager - Properties windows

Toolbar shortcut to access the Tool Manager: 

Opened via pressing the **Properties** button on the Tool Add/Modify/Copy windows, the features on these windows define supporting tool properties, such as: tool color, gage length/offset, control point offset, and more. Properties windows appear different, depending the on the active **Tool Type**. Consult the appropriate section for details.

Tool Properties-Milling window features

Tool Properties-Turning window features

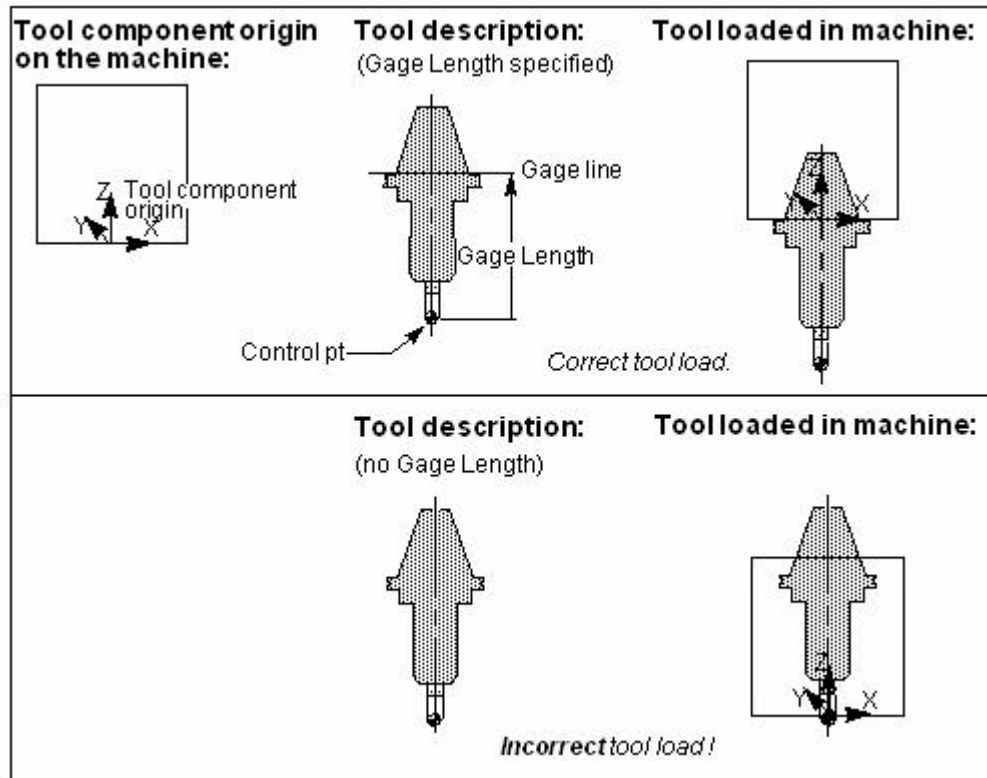
Tool Properties-Milling window features:

Color—Color of the tool assembly. VERICUT uses this color to shade the tool and its cuts when the "**Tool Library**" cut color method is set. (Ref. **Setup menu** > **Motion**) Available colors are as defined on the **File menu** > **Colors: Define** tab.

Gage Length (Z)—Specifies the tool's gage length value. Gage length values are typically positive, as measured from the tool control point (typically the tool tip) to the gage line. Gage lengths are referenced by VERICUT when a G-code tool path is being processed. See "**Programming Method**" on the **Setup menu** > **G-Code** > **Settings** window for more information.

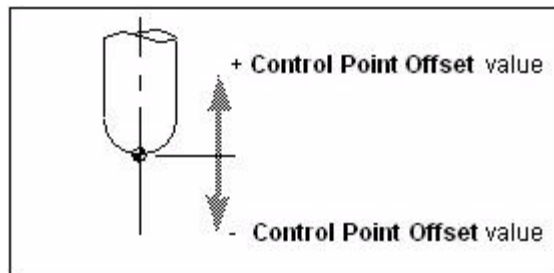
Missing gage length values can cause a tool to load improperly on a machine. (See figure below.)
When defined, the "**Gage Point Offset**" view attribute displays a hollow target symbol at the gage length offset location. (Ref. Tool Manager: **View menu** > **Attributes**)

Sample mill tool gage length applications:



Control Point Offset (Z)—Specifies an offset for the control point, or "driven point". A positive value moves the control point in the positive tool axis direction (closer to the spindle), while a negative value moves the control point in the opposite direction. Material is removed based on motion commands and the tool shape relative to the control point.

When defined, the "**Driven Point Offset**" view attribute displays a hollow target symbol at the gage length offset location. (Ref. Tool Manager: **View menu > Attributes**)



OptiPath Description / #Teeth—Specifies two of the four values needed to identify an optimization record to use. The other two values come from **Material** and **Machine** on the **OptiPath menu > Control** window. If an optimization record that matches the four values is not found, VERICUT continues to write unoptimized tool path records to the optimized tool path file.

Select / Clear—The **OptiPath Description** field is an uneditable text field. Click **Clear** to clear field values, or **Select** to retrieve different values from the OptiPath Library.

Tool Properties-Turning window features:

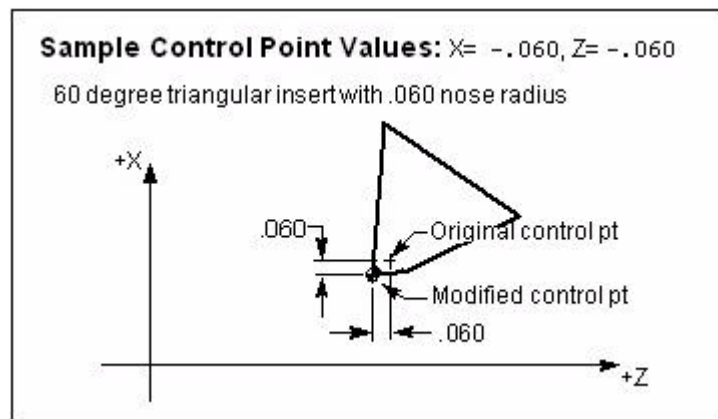
Color—Color of the tool assembly. See "**Color**" for milling tools above for more information.

Gage Offsets (X Y Z)—Specifies the tool's gage offset values, also known as "tool offset". The gage offset values are measured from the tool control point to the tool spindle centerline (X), and gage line (Z). Gage lengths are referenced by VERICUT when a G-code tool path is being processed. See "**Programming Method**" on the **Setup menu > G-Code > Settings** window for more information.

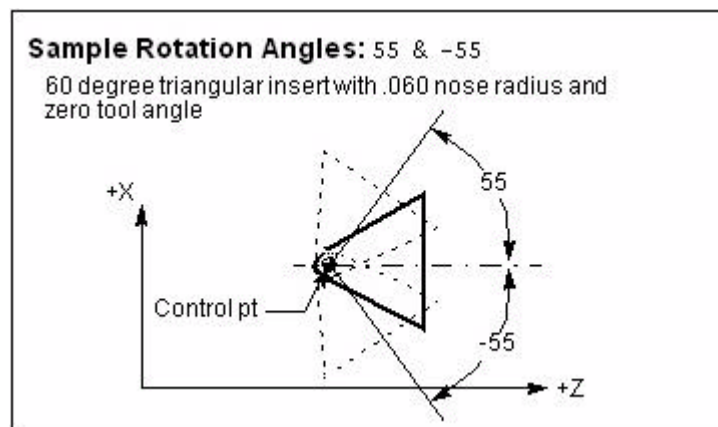
Missing offset values can cause a tool to load improperly on a machine. (See figure below.) When defined, the "**Gage Point Offset**" view attribute displays a hollow target symbol at the gage offset location. (Ref. Tool Manager: **View menu > Attributes**)

Control Point Offset (X Y Z)—Specifies an offset for the control point, or "driven point". Values are entered with respect to the spindle axis (Z), and cross-slide axis (X). Material is removed based on motion commands and the tool shape relative to the control point.

When defined, the "**Driven Point Offset**" view attribute displays a hollow target symbol at the gage length offset location. (Ref. Tool Manager: **View menu > Attributes**)



Rotation Angle—Specifies rotation applied to the turning cutter. Rotation occurs about the tool's control point. The positive rotation direction is from the positive spindle axis towards the positive cross slide axis.



Tool Manager - Position window

Toolbar shortcut to access the Tool Manager: 

Opened via pressing the **Position** button on the Tool Add/Modify/Copy windows, the features on this window move a model in a tool holder by translating or rotating it, or assembling it with others.

Move/Rotate—Configures the window for translating or rotating the tool holder model.

Assemble—Configures the window for assembling the tool holder with others.

The remaining features on this window differ, based on which option above is selected. See the appropriate section for details.

Turning tool position "Move/Rotate" features

Turning tool position "Assemble" features

Turning tool position "Move/Rotate" features

Features on this tab are used to translate or rotate a model in the tool holder. Window features function the same as those described for modeling—see "Move a component or model" in the *Using VERICUT* chapter.

Translate:

From / To / Move—These features translate the tool holder model. Translation is incremental, as calculated from the "**From**" point to the "**To**" point location. Incremental translation is applied each time **Move** is pressed.

XYZ values can be entered (separated by spaces), or filled in by clicking on a model. In general, graphical selection supports picking corner points and midpoints of model geometry. A crosshair and vector show you the pending pick-point location.

Tip: If the applied translation is incorrect, press **Undo** to return the object to its previous location.

Rotate:

Center of Rotation—Specifies XYZ center of rotation used when the rotation buttons are used (see below). XYZ values can be entered (separated by spaces), or filled in by clicking on a model, as described above for **From / To / Move**.



To see the center of rotation, press . To remove the center of rotation symbol press this button again, or close the Position window.

Increment—Specifies incremental degrees of rotation applied when one of the rotation buttons are pressed.

Rotation buttons—(X+/X-, Y+/Y-, Z+/Z-) Applies the incremental degrees of rotation specified in the **Increment** field. Rotation occurs about the **Center of Rotation** relative to the active

component origin.

Tip: If the applied rotation is incorrect, press **Undo** to return the object to its previous location.

Absolute:

Position—Specifies the absolute XYZ position of the object relative to the holder origin. Values in this field are separated by spaces.

Angles—Specifies the absolute XYZ rotation of the object relative to the holder origin. Values in this field are separated by spaces.

Undo—Returns the object to its previous location, or as it was when the Position window was opened.

Turning tool position "Assemble" features

Features on this tab are used to move a model in a tool holder by assembling it with others. Window features function the same as those described for modeling—see "Move a component or model" in the *Using VERICUT* chapter.

Objects are assembled by mating or aligning one to three planar surfaces with surfaces on other models. If a non-planar surface is selected, VERICUT constructs a tangent plane at the pick point. The relationship of surfaces being mated or aligned is known as a "constraint". Constraints must identify a unique location to move object to, or additional constraints must be defined.

Type—Specifies how selected surfaces are constrained (related) to cause object movement. Options are:

- **Mate**—Moves the object so the selected surface opposes the surface selected on the second object (normals oppose each other).
- **Align**—Moves the object so the selected surface is aligned with the surface selected on the second object (normals point in the same direction).

After selecting two surfaces to satisfy a constraint, VERICUT highlights the constraint with a checkmark.

Offset—Distance and direction in which to offset constrained surfaces, as applied normal to the surface.

Undo—Returns the object to its previous location, or as it was when the Position window was opened.

Open (Control file)

Location: **Setup menu > Control > Open**

Toolbar short cut for saving and opening Control files: 

Opens a window to open (load) a Control file. Control files contain data that describes how the NC control processes machine code data.

Save (Control file)

Location: **Setup menu > Control > Save**

Toolbar short cut for saving and opening Control files: 

Saves (updates) an existing Control file with the current NC control settings.

Save As (Control file)

Location: **Setup menu > Control > Save As**

Toolbar short cut for saving and opening Control files: 

Opens a window to save a Control file. Control files contain data that describes how the NC control processes machine code data.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names. To save space, the features described below are unique to VERICUT file selection windows or this window.

Save Control File window features:

Units to Save—Controls the units (Inch or Millimeter) in which to store control configuration data. If the session units are different than the units to save, then values are converted accordingly when stored in the new file.

Word Format

Location: **Setup menu > Control > Word Format**

Opens a window to specify how the NC control interprets G-code words and special characters. (See how)
Once words are defined, they can be "grouped" with address values via the **Setup menu > Control > Word/Address** function to perform specific actions by calling CGTech supplied or custom macros.
Undefined words cause errors when processed, and are not acted on by VERICUT.

Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

Words window features:

Word Name list—List of machine code, or "G-code" data words and characters that are interpreted by the control, arranged in alphabetical order. The list provides a summary of word types, sub-types, inch and metric formats, and multipliers (if used).

Add—Opens a window to add words or special characters to the control configuration. (See window features)

Modify—Opens a window to modify words or special characters in the control configuration. (See window features)

Delete—Deletes the selected word from the control configuration.

Add/Modify Word window features

Opened via pressing **Add** or **Modify** on the Word Format window (see above), the features on this window define how the NC control interprets specific G-code words or special characters. (See how)

Name—Identifies the word or special character. Each word name must be unique.

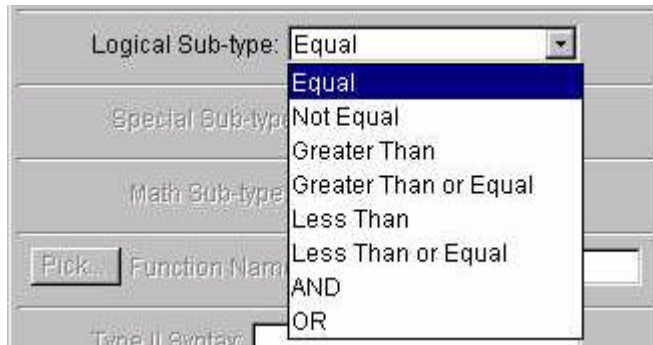
Type—Specifies the type of control function performed by the word. "Sub-types" further differentiate word functions. Available sub-types are determined by the active word **Type**.

Options are:

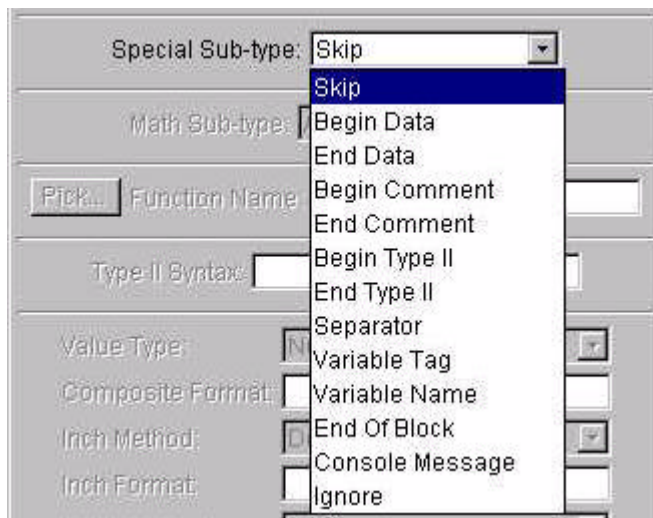
- **Logical**—Evaluates a logical expression. See "**Logical Sub-type**" below for more detailed classifications.
- **Special**—Accesses special NC control functions. See "**Special Sub-type**" below for more detailed classifications.
- **Math**—Performs a mathematical operation. See "**Math Sub-type**" below for more detailed classifications.
- **Function**—Calls the control function specified in the **Function Name** field.
- **Type II**—Specifies that the word is a three character mnemonic representing a Type II data command, e.g. "AXO", "CLS", etc.

- **Macro**—Word will be paired with a value, then configured via the **Setup menu > Control > Word/Address** function to call one or more action macros. (See list of available macros)
- **Conditional**—Similar to the **Functions** word type, except performs multiple functions, depending on the condition of word use. Conditional words are typically supported by special routines written by CGTech or other developers to evaluate word use.

Logical Sub-type—Options for evaluating logical expressions. Example options follow.



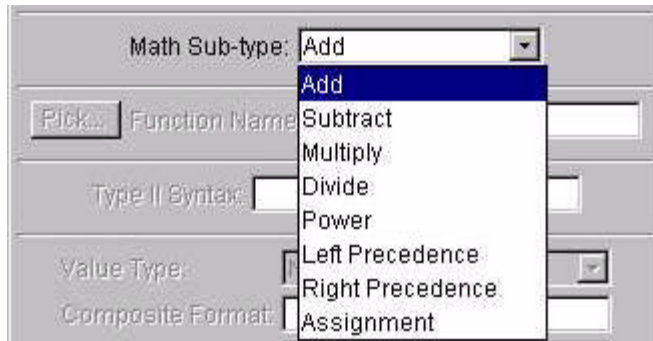
Special Sub-type—Options for performing special control functions. Example options follow.



Notes:

- If a **Begin Data** special word is defined with out specifying an **End Data** word, the **Begin Data** word is used for both.
- **Variable Tag**—Character which identifies the number that follows as a variable register number, e.g. “#”. Example- “#100” indicates variable register number “100” is being referenced.
- **Variable Name**—Alpha-numeric word that identifies a variable which does not have an associated variable register number, e.g. “PPX” (type II variable format).
- **Ignore**—Ignores specific words in the tool path file.

Math Sub-type—Options for performing math operations. Examples follow.



Function Name—Control function called when the word is processed. (See list of available functions)

Type II Syntax—Describes syntax expected in a "Type II" command. Use the descriptors listed below to describe the expected syntax.

- **WV**—word/value pair, e.g. "X1"
- **T**—text string
- **V**—numeric value

Descriptors used in combinations are separated by commas. Examples follow.

Example 1- Configuration for set axis offsets command "(AXO,X1,Y2,Z3)":

- **Word Name=AXO, Type= Type II, Type II Syntax= WV,WV,WV**

Example 2- Configuration for call subroutine command "(CLS,SUB1)":

- **Word Name=CLS, Type= Type II, Type II Syntax= T,V,V,V,V,V**

The "V" descriptors allow passing variables to the subroutine, although none are being passed in the above example call.

Value Type—Specifies how the control interprets values, or "addresses" that follow the word.

Options are:

- **Numeric**—Interprets word values as numbers. The system reads the numeric characters following the macro word until a non-numeric character is found, for example: alphabet, symbol, or separator. The non-numeric character indicates the beginning of a new word.
- **Alphabet**—Interprets word values as alphabet characters. The system reads the alphabet characters following the macro word until a non-alphabetic character is found, for example: number, symbol, or separator.
- **Alpha-Numeric**—Interprets word values as an alpha-numeric text string. The system reads the characters following the macro word until a symbol or separator character is found.
- **Composite-Numeric**—Separates word values into pieces which can be individually acted on. The system uses the **Composite Format** to "parse" the numeric characters following the word until a non-numeric character is found, for example: alphabet, symbol, or separator. See **Composite Format** for an example.

Each separated piece of a composite-numeric word/value is available for calling macros via the

Setup menu > Control > Word/Address function, and can be graphically selected from a word pick list.

- **List-Numeric**—Separates an argument list following the word into pieces which can be individually acted on. This option is useful in a situation like acting on "n" values differently in the sample codes: TEXT(n,n,n,n,...) or TEXT n,n,n,n,... Parens may be present but are not required, and commas separate the arguments. Up to 32 arguments can be listed. Each argument must be numeric, or a math expression which equates to a numeric value.

Similar to **Composite-Numeric** value types, each separated piece of a list-numeric word/value is available for calling macros. However, these pieces do not appear in the word pick list. Specify the pieces via listing the word followed by a space and the sequential number of the argument in the list. Following the above example, you could use the **Setup menu > Control > Word/Address** function to associate macros with "TEXT 1" (corresponds to the first "n" parameter value, "TEXT 2" (second "n" parameter), and so on.

- **None**—Word does not have an associated value. The first character after the word indicates the beginning of a new word.

Examples follow.

Example 1- Assume the word "PGM" is defined as a **Macro** word type. The table below shows how **Value Type** can affect interpretation of the value paired with "PGM".

Sample data block	Value Type=Numeric	Value Type=Alphabet	Value Type=Alpha-Numeric	Value Type=None
PGM123ABC	123	none	123ABC	none
PGMABC123	none	ABC	ABC123	none

Example 2- Consider interpreting the following block of data: "TEST(10,20,30)"

1. In **Word Format**, add word "TEST" as follows:

- **Name=TEST, Type=Macro, Value Type=List-Numeric**

2. In **Word/Address**, add group to act on first argument in TEST list "10":

- **Word=TEST 1, Range=*, Macroname=<desired macro>** (group is passed a value of "10")

3. Add groups to act on second & third arguments in TEST list "20" and "30":-

- **Word=TEST 2, Range=*, Macroname=<desired macro>** (passed value: "20")

- **Word=TEST 3, Range=*, Macroname=<desired macro>** (passed value: "30")

Composite Format—Specifies how to break up, or "parse" **Composite-Numeric** values. Enter one or more numbers separated by spaces to specify the quantity of parsed number values and the significant places of each. An asterisk "*" can be used as a wild card entry. Each parsed value appears in the word pick list of the **Setup menu > Control > Word/Address** function to perform individual actions.

Examples of parsing "T" word/values:

Sample data block	Composite Format	Parsed values

T0203	2 2	T=203, T1=02, T2=03
T102	1 2	T=102, T1=1, T2=02
T0304 or T304	* 2	T=304, T1=03, T2=04
T10001	* 2	T=10001, T1=100, T2=01
T102	1 1 1	T=102, T1=1, T2=0, T3=2
T12345678	2 * 2	T=12345678, T1=12, T2=3456, T3=78
T102.3	* 2.1	T=102.3, T1=1, T2=02.3

Inch Method / Metric Method—Specifies how to interpret inch (and metric) address values. This feature is significant only for values that do not contain a decimal point. Values with a decimal are always interpreted via the "**Decimal**" method.

Options are:

- **Decimal**—Interprets values as floating point decimals.
- **Leading or Decimal**—Interprets values without a decimal as having leading zeros. Depending on the active unit system (inch or metric), values are interpreted as described by the corresponding "Format" (see below).
- **Trailing or Decimal**—Similar to **Leading or Decimal**, except interprets values as having trailing zeros.

Inch Format / Metric Format—Specifies the number of digits before and after the decimal point when interpreting leading or trailing zero values (see above).

Data entry format: a.b

where "a" specifies the number of digits before, and "b" specifies the number of digits after the decimal point.

Multiply / Multiplier—When **Multiply** is active, the word value is multiplied by the amount specified in the **Multiplier** field. (Note that a multiplier of "1" has no effect on the word value.)

Word/Address

Location: **Setup menu > Control > Word/Address**

Opens a window to "group" G-code words defined via **Setup menu > Control > Word Format** with a range of address values, and associate them with action macros to simulate machining center activity. (See how) For simplicity, we'll call each of these associations a "group". Each group can be configured to call one or more action macros to perform various actions. This functionality allows you to virtually "wire" a custom NC control based on machine code information present in the tool path file.

Groups with similar functions are arranged into a "class". (See how) The listed order of classes in the Word/Address window helps determine the timing of when corresponding actions occur.

The Word/Address window dynamically changes, based on items selected in the window. For example, when you select a class, the "Word/Range" list shows the word/address of the groups within the class. The "Condition" list shows conditions defined for the selected group. Finally, the "Macroname" list shows the macros associated with the selected condition in the group. Select a different group to see its defined conditions. Similarly, select a condition to see associated macros.

Class, group, condition, and macro order is changed via dragging them with the right mouse button.

Note: Always reset VERICUT (press  on the VERICUT main window) after making changes to the control configuration.

Word/Address window features:

Class, Word/Range, Condition, MacroName lists—Lists classes, groups, conditions and macros configured in the control.

List order is changed via dragging items with the right mouse button.

Class: Add—Opens a window to specify the name of a class to add. The new class is added after the selected class in the list.

Class: Rename—Opens a window to rename the selected class.

Class: Delete—Deletes the selected class and the word/address groups it contains.

Word/Range list—List of groups defined in a class. This list shows the word name and range of values.

Word/Address: Add—Opens a window to add groups/conditions. (See window features)

Word/Address: Modify—Opens a window to modify groups/conditions in the control configuration. (See window features)

Word/Address: Delete—Deletes the selected group/condition from the control configuration.

Word/Address: Find—Finds groups that match specific G-code word/addresses, or call specific macros. (See window features) Options find the first, next, or previous matching groups in the control configuration.

Add/Modify Word/Address window features

Help on VERICUT

Opened via pressing **Add** or **Modify** on the Word/Address window (see above), the features on this window maintain groups/conditions in the control configuration. (See how)

VERICUT will attempt to add a new group after the group selected in the list. A new group condition is added after the selected condition, and so on. However, if the word and range of an added group matches that of a previously defined group, the new group is automatically added as a condition to the previously defined group. This occurs even when the previously defined group resides in a different class.

Window features:

Word / Variable option—Controls the type of group being defined. Options are:

- **Word, Word/Range**—These features define a group whose actions are performed when a G-code having a corresponding word and address value is processed.

Enter the group word and range of values into the data fields that appear when this option is active. The word must be previously defined via the **Word Format** function. You can type the word character, or select it from a pick list.

The **Range** field commonly specifies a single address. However, you can also specify multiple range values (separated by spaces), inclusive ranges, or use "*" as a wildcard to denote "all values". Examples follow.

Examples of specifying "L" group **Word** and **Range** values:

Group Word	Range	G-codes acted on by the group
L	2	L2
L	1 5	L1, L5
L	1 5-10	L1, L5, L6, L7, L8, L9, L10
L	*	Ln (where "n" is any value)

- **Variable, Variable/Range**—These features define a group whose actions are performed when a specified variable is set to a corresponding value.

Enter the group variable name and range of values into the data fields that appear when this option is active. Variable ranges are specified in the same manner as for words, as described above.

Condition—Controls the type of condition which if met, causes the group to perform different action(s). Options are:

- **Word, Conditional Word/Conditional Range**—Condition based on another word and address value range appearing in the G-code data block. This is the most common condition type. Enter the word and range of values with the same considerations as described above for **Word, Word/Range**.
- **State, Conditional State/Conditional Value**—Condition based on a machine state. Choose the desired state and value from the pick list.
- **Variable, Conditional Variable/Conditional Range**—Condition based on when a variable is set to a corresponding value. Enter the variable name and range of values with the same considerations as described above for **Variable, Variable/Range**.

Macroname/Variable option—Controls if group calls a macro, or sets a variable. Depending on the active choice, features are provided for specifying the macro called (select or type in field), or

the name of the variable to be set (enter the variable value in the **Override Value** field).

Hint: An automatic filtering capability assists you with finding macros. When you enter text characters in the **Macroname** field, the list of macros is automatically filtered to show those that match the specified text. Text entered is not case sensitive.

Process during Scan—When active, calls the macro during a "scan pass" prior to tool path processing (ref. "Scan Toolpath Files" on the **Setup menu > G-Code > Settings** window **Settings** tab). The default condition, is to call the macro during tool path processing.

Process After Motion—When active, performs the group action after motion commands in the data block have been processed. The default condition is to perform the group action according to the rules of normal G-code data processing. (Ref. "About building NC controls" in the *Using VERICUT* chapter.)

Override Value—This feature acts differently, depending on if **Macroname** or **Variable** is selected:

- With **Macroname** selected—Specifies a value to pass to the specified macro. If blank, the address value accompanying the word is passed.
- With **Variable** selected—Specifies a value to assign to the specified variable. Math expressions and variables understood by the control are supported. Enter "\$" to use the address value accompanying the word. If the override expression does not contain "\$" or variable names, it is evaluated immediately and only the value is retained. If the expression can not be evaluated immediately, the full expression is retained in the control configuration and evaluated separately for each block that activates the group.

Examples of using **Override Value**: (given variable #2=3)

Text entered in Override Value field	What is retained in control configuration	Sample G-code data block	Resulting value when processed
SIN(30)	.5	X5	.5 (constant value- not affected by G-code data)
SIN(30)	.5	X10	.5 (see note above)
\$*10	\$*10	X5	50
(\$+#2)*10	(\$+#2)*10	X5	80

Override Text—Similar to **Override Value**, except specifies text to pass to the macro.

Note: Only certain macros are designed to accept text values. Consult the section titled "Macros - listed alphabetically" in the *Reference* chapter for more information.

Find Word/Address window features

Opened via pressing **Find** on the Word/Address window (see above), the features on this window help you find the groups in a control configuration that are associated with a specific word/address value, or call specific macros.

Words/Range tab—Finds groups associated based on a specific word/address value. Pick the desired word from the **Word list**, then enter the address value in the **Range** field.

Macroname tab—Finds groups that call a specific macro. Type or pick the name of the desired

Help on VERICUT

macro from the **Macroname list**.

After configuring the window to find the desired group, use the buttons at the bottom of the window to find the **First**, **Next**, or **Previous** occurrence in the control configuration.

MDI

Location: **Setup menu > Control > MDI**

Toolbar short cut: 

Opens a window to manually enter and process blocks of G-code data. The MDI, or "Manual Data Input" function provides a quick and easy way of verifying that the machine/control combination responds to G-code data commands as expected.

Notes:

For machine motions to be seen, **Machine Simulation On** must be active- ref. **Setup menu > Machine > Settings**.

Calls to subroutines and NC macros are not supported in the MDI window. Use a short test tool path file instead.

Do not attempt to use MDI before machine construction is complete. If the Stock, Tool, and all motion components are not present in the machine definition, the machine may not respond to motion commands as expected.

After adding motion type components to a machine, always press **Reset Model**  before using MDI to test machine movements.

General (Control settings)

Location: **Setup menu > Control > Settings > General**

Opens a window to configure general guidelines for how the control processes machine code data and number equations. (See how)

General control settings window features:

Default Machine Mode—Sets the default machine mode for multi-operational machine, such as a mill-turn machining center. Options are: **Milling**, **Turning**, or **Wire EDM**.

Control Type—Sets the type of NC control being used. Choosing the proper option establishes a mode of operation consistent with how the control processes NC data. Options are:

-**Generic**—Fanuc and most other controls using standard controller functions.

-**NUM**—French-made control. This option affects how G-code data is interpreted, for example: how variables are initialized and processed, and how to interpret parameters to macros that shift the location of the tool path.

-**Heidenhain Conversational**—Heidenhain Conversational control. This option causes VERICUT to recognize the "L" word as an optimizable word. Also, the "L" word is included with any cuts added during optimization.

Calculation Tolerance—Tolerance for rounding mathematical evaluations, such as determining if two calculated values are equal to, greater than, or lesser than each other.

Order of Math Operations—Order in which math operations are performed. Options are:

- **Rules of Precedence**—Follow the basic rules of math precedence:

1. Perform exponential (power) operations
2. Perform multiplication and division operations
3. Perform addition and subtraction operations

Example: $5 + 5.3 * 3 * \sin(30) = 12.95$

Enclosing math operations in parenthesis causes them to be performed before those not enclosed in parenthesis. The same rules of precedence are applied to math operations enclosed in parenthesis.

- **Left to Right**—Use left to right sequence.

Example: $5 + 5.3 * 3 * \sin(30) = 15.45$

Default Word—Word assumed by VERICUT for NC data blocks beginning with numbers, such as NC data for some Heidenhain controls. For example, for VERICUT to interpret blocks like "50G01X5", set **Default Word** to "N". VERICUT then interprets the block without error as "N50G01X5".

Motion (Control settings)

Location: **Setup menu > Control > Settings > Motion**

Opens a window to configure default NC control motion states, and the precision used when outputting calculated values. (See how) These settings are in affect when the control and machine are initially powered on.

Motion control settings window features:

Default Motion Type—Default motion type for the control. Options are: **Rapid** or **Linear**.

Default Plane Selection—Default motion plane, or "cutting plane" for the control. Options are: **XY**, **ZX** or **YZ**.

Default Control Units—Default measurement units for the control. Options are: **Inch** or **Metric**.

Default Command Mode—Default command mode, or "input dimension mode" for the control. Options are: **Absolute** or **Incremental**.

Default Feed Mode / Default Feedrate—Default feed rate mode and value for the control. Feed mode options are: **Feed per Minute** or **Feed per Revolution**.

Output Initial Spindle Location—When active, causes the machine to move to its initial spindle location at the beginning of tool path processing. Enter initial machine location values in an Initial Machine Location table.

Rapid Motion Cancels Cycles—When active, rapid motions (e.g. G0) cancel canned cycles (e.g. G81-89).

Linear Motion Cancels Cycles—When active, linear motions (e.g. G01) cancel canned cycles (e.g. G81-89).

CW Motion Cancels Cycles / CCW Motion Cancels Cycles—When active, clockwise (or counterclockwise) circular interpolation motions (e.g. G02) cancel canned cycles (e.g. G81-89).

Output Rapid Motions as—Controls the format in which rapid motions are written to the APT Output file during revers post-processing. Options are:

- **Rapid**—Output the word "RAPID" followed by a single tool tip location.
- **Feedrate**—Output "FEDRAT/n" where the feed rate is calculated based on the rapid priority of moving axes, then output the tool tip location(s). When rapid priority causes axes to move independently, multiple tool tip positions are output.

Maximum Inverse Time Output Feedrate—Highest feedrate value which can result from interpreting inverse time feed rate values (e.g. G93). The output feed rate is set to this rate when in inverse time feed mode and the calculated feed rate is zero or greater then the specified maximum.

Output Precision—Controls the accuracy used to interpret calculated values, such as: CDC offset positions, mathematical operations, etc. Options are:

- **Full Precision**—Highest possible accuracy. Note that full precision output values can result in higher accuracy than the NC control uses, which may result in differences between simulated motion and the actual motion performed by the NC control.

Help on VERICUT

- **Input Precision**—Use the accuracy used in the input tool path file. For example, when tool path file data is in 3.4 format, the precision used is also 3.4.
- **Specified Precision**—Use the accuracy specified via **Specified Output Precision** (see below).

Specified Output Precision—When **Output Precision**= **Specified Precision**, this value reflects the accuracy used by the NC control to perform the operations described above for **Output Precision**.

Circles (Control settings)

Location: **Setup menu > Control > Settings > Circles**

Opens a window to configure how the circle center data (e.g. IJK) is interpreted. (See how)

Note: The **Interpolation Tolerance** determines the quantity of machine positions which simulate arc and helical motions. (Ref. **File menu > Properties: Tolerance** tab)

Circles control settings window features:

Circle Center Mode—Mode in which to interpret circle center data. Options are: **Absolute** (absolute center coordinates), **Incremental** (incremental distances-see "**Incremental Circle Center Method**" below), or **G Code Dependent** (as previously described, depending on the active input dimension mode: absolute or incremental).

Incremental Circle Center Method—When interpreting incremental circle center values, controls how incremental circle center data is interpreted. Options are: **From Start to Center Point** (distance from circle start point to circle center), **From Center to Start Point** (distance from circle center to circle start point), or **Unsigned** (unsigned, positive incremental distance between circle start point and circle center).

Note: Unsigned incremental circle motion blocks are always positive in value, and do not cut circles greater than 90 degrees.

Pitch Specified with IJK—When active, helical motion pitch is controlled by circle center data. Clear this checkbox to indicate pitch is controlled by other data, such as Zn.

APT Output for Circular Motion—Controls the format in which circular motions are output during conversion to an APT output file. An APT Output file is created when **Create APT Output File** is active. (Ref. **Setup menu > G-Code > Process Options > APT Output**)

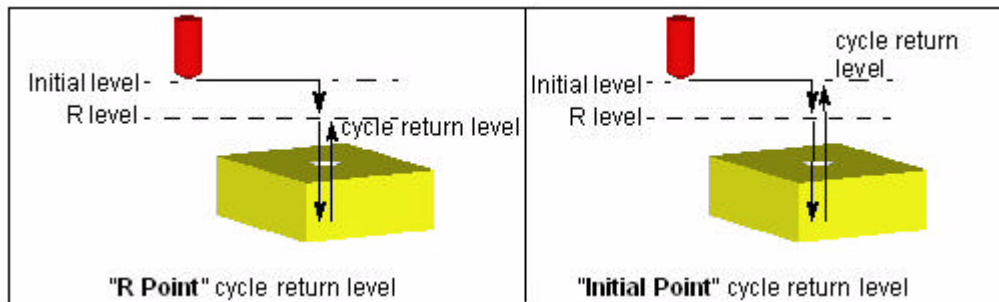
Cycles (Control settings)

Location: **Setup menu > Control > Settings > Cycles**

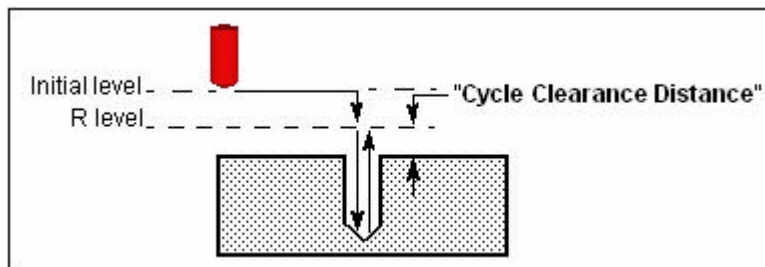
Opens a window to configure how fixed tool axis, or "canned" cycle motion blocks are interpreted, e.g. G8n. (See how)

Cycles control settings window features:

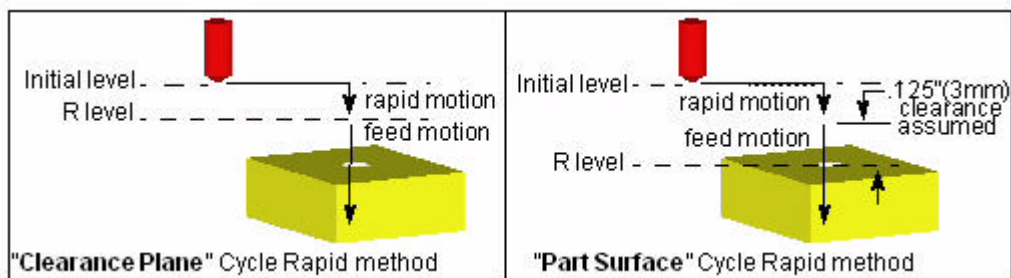
Cycle Return Level—Controls where the tool tip is returned (retracted) after processing a cycle command. Options are: **R Point** (rapid positioning or "R level"), **Initial Point** (initial level from where the cycle motion began), **Specified Point** (clearance distance from the workpiece as specified by an word/address, such as G71Zn with some Okuma controls), or **Clearance + Retraction** (similar to **Specified Point**, except a retraction clearance can be specified).



Cycle Clearance Distance—Distance from the workpiece to the R level.



Cycle Rapid method—Controls how rapid positioning is performed between cycle executions. Options are: **Clearance Plane** (FANUC method-R level located at some clearance above the workpiece, tool tip placed at the R level), or **Part Surface** (Cincinnati Acramatic method-R level located on the workpiece surface, tool tip placed .125", or 3mm above the R level).



Cycle Depth Value—Controls how the cycle depth values are interpreted. Options are: **Absolute** (absolute cycle depth value), **Incremental** (incremental distance-see "**Cycle Incremental Depth Value**" below), or **G Code Dependent** (as previously described, depending on the active input dimension mode: absolute or incremental).

Cycle Incremental Depth Value—When interpreting incremental cycle depth values, controls how incremental distances are interpreted. Options are: **Relative to Cycle Rapid Value** (incremental distance from R level), **Relative to Cycle Initial Level** (incremental distance from initial level), or **Relative to Part Surface** (incremental distance from the part surface).

Cycle Rapid Value—Controls how cycle rapid values are interpreted to determine the R level. Options are: **Absolute** (absolute R level position), **Incremental** (incremental distance to the R level- see "**Cycle Incremental Rapid Value**" below), or **G Code Dependent** (as previously described, depending on the active input dimension mode: absolute or incremental).

Cycle Incremental Rapid Value—When interpreting incremental cycle rapid values, controls how incremental distances are interpreted to determine the R level. Options are: **Relative to Cycle Initial Level** (incremental distance from the initial level), or **Relative to Part Surface** (incremental distance from the part surface).

Cycle Causes Rapid Motion—When active, sets the rapid motion mode with a cancel cycle command (e.g. G80).

Ignore Cycles during Rotary Motion—When active, ignores canned cycles during rotary motions. Clear this checkbox to allow cycles during rotary motions.

The use of this modal becomes more important when the **Output Intermediate Points** rotary motion setting is in use. (Ref. **Setup menu** > **Control** > **Settings** > **Rotary**)

Cycle Execute—Controls when cycle motions are executed. Options are: **On Motion** (execute cycle on cycle block and each following motion until cancelled), or **As Commanded** (cycle definition block is for setup only; execute cycle when commanded by a specific word/address, such as G79 with some Phillips NC controls).

Tooling (Control settings)

Location: **Setup menu > Control > Settings > Tooling**

Opens a window to configure default tooling conditions and tool change activity. (See how)

Tooling control settings window features:

Initial Tool Component ID—Specifies the **Tool Index Number** of the tool component initially active for tool changes.

Tool Number Method—Controls how to interpret tool number word/addresses (e.g. Tn). Options are:

- **Select Only**—Tool number only selects the tool. The tool is changed by a separate command, e.g. M6)
- **Select & Change**—Tool number selects and changes the tool. A separate tool change command is not used.

Tool Change Retract Method—Method of retracting for a tool change. Methods which retract machine axes reference location values stored in a Tool Change Location table. Options are:

- **No Retract**—Change the tool at its current location.
- **Retract (Z-Axis only)**—Retract the tool only along the Z-axis.
- **Retract All Axes**—Retract all machine motion axes to their respective tool change locations.
- **Retract Tool Side Axes**—Similar to **Retract All Axes**, except applies only to motion axes connected between the machine Base and Tool components.
- **Use Retraction Table**—Refer to the Tool Change Retraction table to determine which machine axes to retract.

Tool Change Cancels Cycles—When active, cancels canned cycles with a tool change.

Tool Change Causes Rapid Motion—When active, sets the rapid motion mode with a tool change.

Rotary (Control settings)

Location: **Setup menu > Control > Settings > Rotary**

Opens a window to configure how rotary motion commands are interpreted, e.g. A, B, C. (See how) A "moving tool philosophy" is assumed when describing rotations.

Rotary control settings window features:

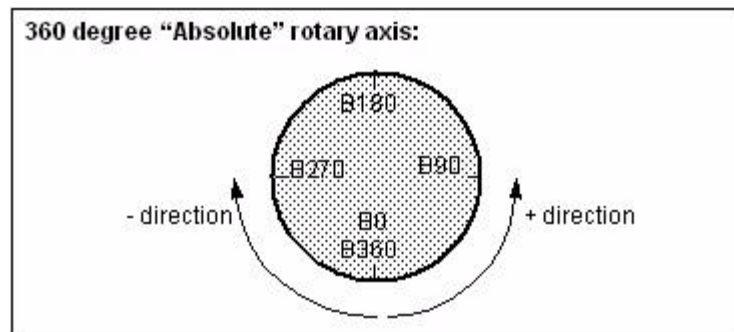
Output Intermediate Points—When active, generates intermediate tool positions during rotary motions. If turned off, only the final tool position of a rotary motion is shown.

Rotary intermediate points help you see the tool-to-workpiece relationship during rotary motions, however, the simulation will be slower. The **Interpolation Tolerance** determines the quantity of machine positions which simulate rotary motions. (Ref. **File menu > Properties: Tolerance** tab)

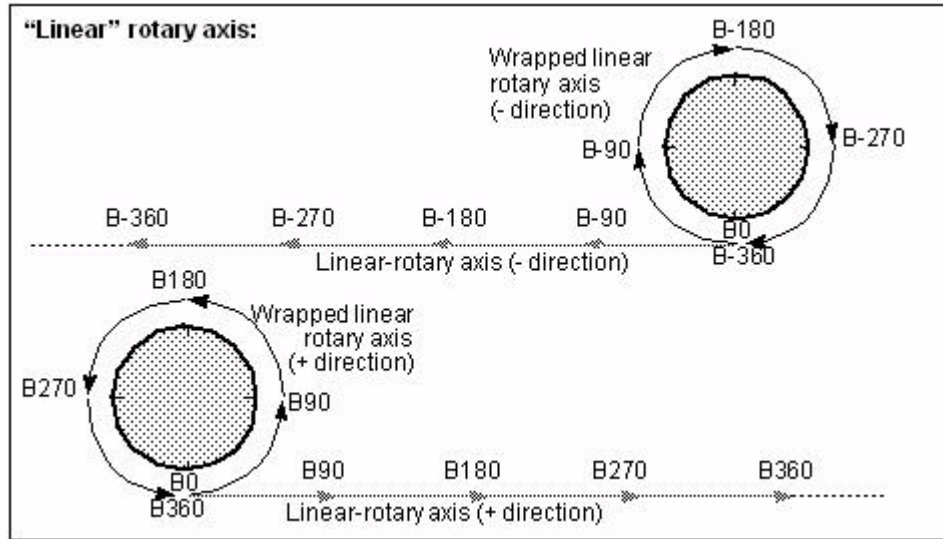
OptiPath Note: Multi-axis motions are not optimized when **Output Intermediate Points** is active.

A-axis Rotary Type, B-axis Rotary Type, C-axis Rotary Type—The settings control how rotary commands are interpreted (e.g. ABC). Each axis is individually controlled. Options are:

- **EIA (360 Absolute)**—Refers to absolute angle positions. **Absolute Rotary Direction** controls how rotation values determine the direction of rotation.



- **Linear**—Refers to angles on a linear axis "wrapped" around the rotary component. In absolute input dimension mode (e.g. G90), rotary values specify absolute locations along the linear-rotary axis, while the sign (+ / -) controls which end of the linear axis is used. In incremental input dimension mode (e.g. G91), rotary values specify degrees to rotate from the current position, and the sign controls the direction of rotation: plus = CCW, minus = CW.



Example rotations of a B-axis linear-rotary component:

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
B0	G90B75	75 deg CCW	B75
B0	G90B-75	75 deg CW	B285 (B-75)
B35	G90B75	40 deg CCW	B75
B35	G90B-75	110 deg CW	B285 (B-75)
B150	G90B120	30 deg CW	B120
B150	G90B-120	270 deg CW	B240 (B-120)
N/A	G91B75	75 deg CCW	Current +75
N/A	G91B-75	75 deg CW	Current -75

Absolute Rotary Direction—Controls how rotation values determine the direction of rotation for **EIA (360 Absolute)** rotary type components. Options and examples follow.

- **Positive -> CCW**—Sign controls the direction of rotation: plus = CCW, minus = CW, and absolute value of the rotation value specifies the rotary destination. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirPosCCW* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Positive -> CCW**:

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	20 deg CCW	B20

B20	G90B-90	290 deg CW	B90
B90	G90B60	330 deg CCW	B60

- **Positive -> CW**—Similar to **Positive -> CCW**, except the rotation directions are reversed. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirPosCW* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Positive -> CW:**

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	340 deg CW	B20
B20	G90B-90	70 deg CCW	B90
B90	G90B60	30 deg CW	B60

- **Always CCW**—Rounds the rotation value to increments of 360, then subtracts 360 until the subtracted value is between 0-360. The rounded value specifies the rotary destination. The tool always rotates in a CCW direction about the rotary center point. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirCCW* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Always CCW:**

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	20 deg CCW	B20
B20	G90B-90	250 deg CCW	B270
B270	G90B60	150 deg CCW	B60

- **Always CW**—Similar to **Always CCW**, except the rotation directions are reversed. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirCW* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Always CW:**

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	340 deg CW	B20
B20	G90B-90	110 deg CW	B270
B270	G90B60	210 deg CW	B60

- **Shortest Distance**—Uses the rotation value and its sign to specify the absolute rotary position. For each rotary motion, the tool rotates the shortest distance about the rotary center point to the rotary destination. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirShortestDist* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Shortest Distance:**

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	20 deg CCW	B20
B20	G90B-90	110 deg CW	B270
B270	G90B60	150 deg CCW	B60

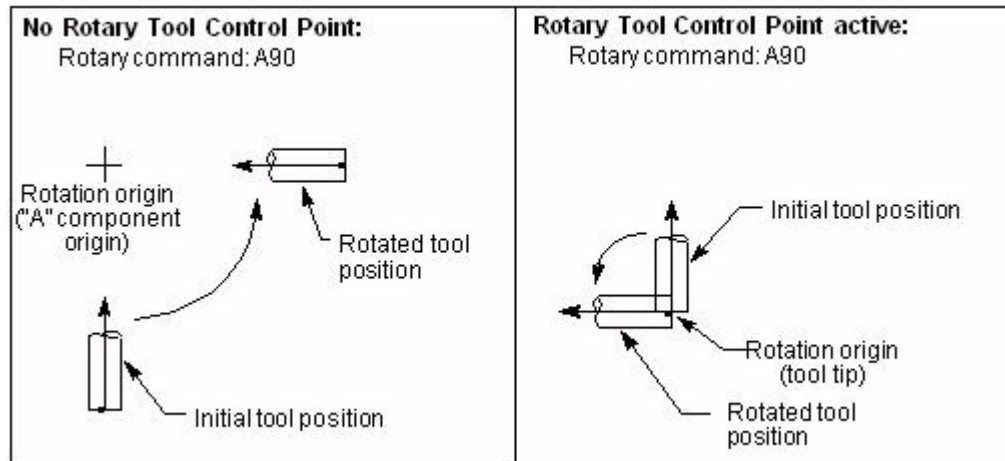
- **Linear**—Rounds the rotation value to increments of 360, then subtracts 360 until the subtracted value is between 0-360. The difference between the rounded value and the current position is the rotary destination while the sign of the difference specifies the rotation direction. A positive difference value rotates the tool in a CCW direction about the rotary center point, a negative value rotates the tool in a CW direction. **Note:** You can configure the same behavior using **Setup menu > Control > Word/Address** and associating the *RotaryDirLinear* macro.

Examples of rotary behavior with **Absolute Rotary Direction=Linear:**

Current axis position	Command	Rotary motion (moving tool philosophy)	Rotary destination
NA	G90B0	NA	B0
B0	G90B380	20 deg CCW	B20
B20	G90B-90	250 deg CCW	B270
B270	G90B60	210 deg CW	B60

Rotary Tool Control Point—When active, Rotary Tool Control Point, or “RTCP” causes rotary motion about a specific control point-typically the tool tip. When RTCP is not active, rotation occurs about the rotary component origins. (Intended for use only with tilting head machines.)

Affect of using RTCP with an A-axis rotary component:



Notes:

- When RTCP is used with machines having multiple rotary axes between the Tool and Base components, the axis of rotation for all rotating components must intersect for RTCP to function as designed. The rotary component origins do not need to be coincident.
- See also "RTCP Pivot Offset table" in the *Reference* chapter for possible additional configuration for correct RTCP simulation.

WireEDM (Control settings)

Location: **Setup menu > Control > Settings > Wire EDM**

Opens a window to configure wire EDM machining settings. (See how)

Wire EDM control settings window features:

Maximum Wire Angle—Maximum angle the wire can tilt in any direction from vertical alignment. If the wire is tilted beyond this angle, an error message is issued.

Initial Voltage—When active, indicates voltage to the wire is on at the start of tool path processing.

UV Relative to XY—When active, U and V tool path values are relative to the current X and Y axis positions.

XY Also Moves UV—When active, X and Y tool path values also cause the corresponding U and V axis to move.

In typical wire EDM machine construction, the U and V axes are connected such that they are slaved off the X and Y axes. In other words, when the X and Y axes move, the corresponding U and V axes also move by default. The **XY Also Moves UV** setting is intended to work with machines built such that the U and V axes are separate from the X and Y axes.

Work Table to XY Output Plane—Distance from the top of the work table (machine Z zero) to the XY output plane (output Z values). The XY output plane is the Z-axis level at which X and Y values are assumed to be driven. This value is measured along the machine Z-axis, and is usually zero.

Cutter Compensation (Control settings)

Location: **Setup menu > Control > Settings > Cutter Compensation**

Opens a window to configure how the programmed tool path is compensated when cutter diameter compensation, or "CDC" is used, e.g. G41-42. (See how)

CDC offset values must be entered into a Cutter Compensation table. When CDC is active, the programmed tool path is simulated just as it will be compensated by the NC control. (Ref. **Setup menu > G-Code > Settings: Settings** tab)

Cutter Compensation control settings window features:

CDC ON/OFF Method—Controls how the offset condition is established when CDC is turned on, and how the offset is cancelled when CDC is turned off. Options are:

- **Ramp ON/OFF**—Establish the offset condition with a “ramp-on” motion after CDC is turned on. When CDC is turned off, the offset is cancelled with a “ramp-off” motion on the next tool move.
- **Immediate**—Immediately establishes the offset condition when CDC is turned on, and assumes the offset has been established by a prior data block (e.g. G43-44 as interpreted by the Phillips CNC controls). When CDC is turned off, the offset is cancelled without a tool move.

CDC Ramp ON/OFF with—Specifies the motion types which establish/cancel the CDC offset. Options are: **Active Plane Motion** (motion in the active cutting plane), or **Any Motion** (any motion, including motion along the tool axis).

Apply Tool Nose Comp with CDC—When active, Tool Nose Compensation table values are used to offset a turning tool path. Tool nose compensation applies especially to tool paths programmed using the Fanuc style “imaginary tool nose programming” method.

Offsets (Control settings)

Location: **Setup menu > Control > Settings > Offsets**

Opens a window to configure an initial work offset, or “fixture offset”, to be in effect at the start of tool path processing. (See how)

Offsets control settings window features:

Initial Work Offset—When active, sets the initial work offset specified by the **Initial Work Index** (see below).

Initial Work Index—Index (offset register value) containing the initial work offset. Offset remains effective until a different work offset (e.g. G54-59) is processed, or until cancelled. Enter the corresponding offset values into a Work Offsets table .

Subroutines (Control settings)

Location: **Setup menu > Control > Settings > Subroutines**

Opens a window to configure how subroutine names are referenced by the NC control. (See how)

Subroutines control settings window features:

Type of Subroutine Names—Controls how names of subroutines are referenced. options are:

- **Numeric**—Use the numeric value; leading zeros are ignored. For example, “O0010” is interpreted the same as “O10”.
- **Text**—Use the text string value. For example, “O0010” is not interpreted the same as the name “O10”.

OptiPath (Control settings)

Location: **Setup menu > Control > Settings > OptiPath**

Opens a window to configure settings referenced when G-code tool path files are optimized by OptiPath. (See how)

The settings are used to control block sequencing and word/value spacing in the optimized output tool path file. Note that by default, optimized blocks receive the same sequence number, if any, that appears on the original block processed. This typically results in multiple blocks having the same sequence number.

OptiPath control settings window features:

Auto Sequence Numbering— When active, re-sequences optimized tool path records in ascending order. The **Sequence Increment Value** (see below) is used for the beginning sequence number, and as the increment value for subsequent sequence numbers.

Both optimized and non-optimized blocks with sequence numbers are re-sequenced. Sequence numbers are not added to unoptimized blocks without a sequence number.

Note: If the **Setup menu > Control > Settings > General: Default Word**="N", then "N" does not appear in the sequence number. The result is a sequence number without a word, for example: "50G01X5", such as required by Heidenhain controls.

Sequence Increment Value—Initial value and increment value for re-sequencing optimized G-code tool path records.

Unoptimized Tool Path:	Optimized Tool Path 1: (default sequencing)	Optimized Tool Path 2: (Auto Sequence Numbering, Sequence Increment Value=10)
N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30G01X2.0Y3.2Z1.0F10.0 N40Y-1.0 N50X2.5 N60Y7.0 N70G91Z2 N80G90G00X-1.0Y-1.0Z3.0	N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30G01X2.3Y3.2Z2.2F150. N30X2.25Z2.F25. N30X2.2Z1.8F15. N30X2.1Z1.4F10. N30X2.Z1.F5. N40Y0.F15. N40Y-0.4F45. N40Y-1.F150. N50X2.5 N60Y-0.6 N60Y-0.4F35. N60Y3.F30. N60Y3.2F25. N60Y6.F20. N60Y6.2F60. N60Y6.6F45. N60Y7.F150. N70F10. N70G91Z2 N80G90G00X-1.0Y-1.0Z3.0	N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30G01X2.3Y3.2Z2.2F150. N40X2.25Z2.F25. N50X2.2Z1.8F15. N60X2.1Z1.4F10. N70X2.Z1.F5. N80Y0.F15. N90Y-0.4F45. N100Y-1.F150. N110X2.5 N120Y-0.6 N130Y-0.4F35. N140Y3.F30. N150Y3.2F25. N160Y6.F20. N170Y6.2F60. N180Y6.6F45. N190Y7.F150. N200F10. N210G91Z2 N220G90G00X-1.0Y-1.0Z3.0
	Ascending block sequence numbers by "10s"	
Duplicate block sequence numbers		

Word Value Spacing—When active, includes a space between word/value pairs in the optimized tool path file. Unoptimized records do not receive spacing. When this feature is not used, optimized tool path records are not spaced.

Unoptimized Tool Path:	Optimized Tool Path 1: (default spacing)	Optimized Tool Path 2: (Word Value Spacing on optimized records)
N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30G01X2.0Y3.2Z1.0F10.0 N40Y-1.0 N50X2.5 N60Y7.0 N70G91Z2 N80G90G00X-1.0Y-1.0Z3.0	N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30G01X2.3Y3.2Z2.2F150. N30X2.25Z2.F25. N30X2.2Z1.8F15. N30X2.1Z1.4F10. N30X2.Z1.F5. N40Y0.F15. N40Y-0.4F45. N40Y-1.F150. N50X2.5 N60Y-0.6 N60Y-0.4F35. N60Y3.F30. N60Y3.2F25. N60Y6.F20. N60Y6.2F60. N60Y6.6F45. N60Y7.F150. N70F10. N70G91Z2 N80G90G00X-1.0Y-1.0Z3.0	N10T1M6 N20G00X2.5Y3.2S500M03MD8 N30 G01 X2.3 Y3.2 Z2.2 F150. N30 X2.25 Z2. F25. N30 X2.2 Z1.8 F15. N30 X2.1 Z1.4 F10. N30 X2. Z1. F5. N40 Y0. F15. N40 Y-0.4 F45. N40 Y-1. F150. N50 X2.5 N60 Y-0.6 N60 Y-0.4 F35. N60 Y3. F30. N60 Y3.2 F25. N60 Y6. F20. N60 Y6.2 F60. N60 Y6.6 F45. N60 Y7. F150. N70 F10. N70G91Z2 N80G90G00X-1.0Y-1.0Z3.0
No word/value spacing since unoptimized tool path did not contain spaces	Spacing (on optimized records only)	

User Defined (Control settings)

Location: **Setup menu > Control > Settings > User Defined**

Opens a window to configure user defined control settings supplied via a custom CME file. This window will appear differently, depending on the CME file in use.

CME files are created using the Machine Developer's Kit, and specified for use in VERICUT via **Dev Kit CME** in the **Setup menu > Control > Adv. Options** function.

Note: Developer's Kit-defined settings can also appear under **Setup menu > G-Code > Process Options > User Defined**, as determined by the developer.

Adv. Options

Location: **Setup menu > Control > Adv. Options**

Opens a window with advanced NC control capabilities, such as: specify subroutines available in the NC control, perform actions at key processing events (e.g. start of tool path processing, start processing a block, etc.), substitute text strings, and more.

Advanced Control Options window features:

- Subroutines tab features

- Events tab features

- Substitute tab features

- Dev Kit CME tab features

Advanced Control Options "Subroutines" tab features

The features on this tab specify names of external files containing subroutines that can be accessed by the control. For additional information about subroutines, see one of the following topics:

What do you want to do?

- Learn about simulating subroutines

- See how to add, modify, or delete subroutine file names

- Configure "Subroutines" control

Advanced Control Options "Events" tab features

The features on this tab maintain macros called and variables set based on key events that occur during tool path file processing. (See how) Event driven macro calls/variable assignments are configured in a similar fashion as in the **Setup menu > Control > Word/Address** window-please review that section first.

The Events window displays the list of macros and variable assignments performed for that event. When a different event is chosen, the window shows the list of macros and variable assignments for that event. Macro/variable order is changed via dragging them with the right mouse button.

IMPORTANT! The macro events listed below are required for G-code processing and should NOT be disturbed.

- Start of Processing => *ConversionInit*

- End of Processing => *ConversionFinish*

- Start of Block Processing => *BlockInit*

End of Block Processing => *BlockFinish*

Events tab features:

Event—Controls which event will call a macro or set a variable. Available events are listed below, most of which are self-explanatory.

- **Reset**—Event occurs before the tool path file is scanned.
- **Start of Processing**—Event occurs after the tool path file is scanned, but before the tool path file is processed.
- **End of Processing**
- **Start of Block Processing**
- **End of Block Processing**
- **Start of Type II Command**
- **End of Type II Command**

Macro/Variable list—Lists the macro/variable assignments for the current **Event**. When a different **Event** is selected from the option list, the Macro/Variable list shows the macro/variable assignments for that event.

Add / Modify—Opens windows to add (or modify) an event-based macro calls/variable assignments. Features in this window behave similar to those described for word/address groups under "Add/Modify Word/Address window features" in the section for **Setup menu > Control > Word/Address**.

Delete—Deletes the selected event-based macro call/variable assignment.

Advanced Control Options "Substitute" tab features

The features on this tab maintain text strings that are substituted when specific text in the tool path file is encountered. (See how) All substitutions occur prior to parsing the words and values in the G-code data block. Using substitutions lets you to alter how VERICUT "sees" data in a block without editing the G-code tool path file.

Substitution (Input/Output Text) list—Lists text string substitutions that will occur when specified input text is encountered.

Input Text / Output Text—Specifies an input text string to substitute, and the output text that VERICUT will use in its place.

Add—Adds a new substitution to the list.

Modify—Modifies an existing substitution in the list.

Delete—Deletes the selected substitution from the list.

Advanced Control Options "Dev Kit CME" tab features

Help on VERICUT

Dev Kit CME File—Opens a window to specify the CME file, or CGTech Macro Executable file containing custom macros that assist the NC control with processing machine code data. This file can also supply "control settings" that allow users to further customize how machine code data is interpreted, and provide access to custom data tables for storing job, tool, and machine related data.

A custom CME file is typically created by CGTech or other services organization using the Machine Developer's Kit. In this manner, end users can customize VERICUT's ability to simulate machine code data according to site-specific needs.

For more information about developing customer CME files, see the on-line documentation for using the Machine Developer's Kit in the CGTech On-line Help Library.

Open (Machine file)

Location: **Setup menu > Machine > Open**

Toolbar short cut for saving and opening Machine files:



Opens a window to open (load) a Machine file. Machine files contain data that describes NC machine construction, kinematics, and other properties.

Save (Machine file)

Location: **Setup menu > Machine > Save**

Toolbar short cut for saving and opening Machine files: 

Saves (updates) an existing Machine file with the current NC machine settings.

Save As (Machine file)

Location: **Setup menu > Machine > Save As**

Toolbar short cut for saving and opening Machine files: 

Opens a window to save a Machine file. Machine files contain data that describes NC machine construction, kinematics, and other properties.

Most features on this window are standard file selection window features that enable you to navigate through directories, filter files, and type or select file names. To save space, the features described below are unique to VERICUT file selection windows or this window.

Save Machine File window features:

Units to Save—Controls the units (Inch or Millimeter) in which to store machine configuration data. If the session units are different than the units to save, then values are converted accordingly when stored in the new file.

Settings (Machine)

Location: **Setup menu > Machine > Settings**

Opens a window to configure settings for an NC machine, such as: collision checking, travel limits, axis priority (for rapid motion), location tables, and when machine motions are simulated.

Machine Settings window features:

Machine Simulation On—When active, simulates machine tool motions when a 3-D machine is displayed in a machine view.

Machine Up—Specifies which way is "up" (i.e. towards the ceiling) for a machine when a machine view displays walls in the background. (Ref. **View menu > Attributes: Background**). The up direction is relative to the machine origin.

Collision Detect tab features

Tables tab features

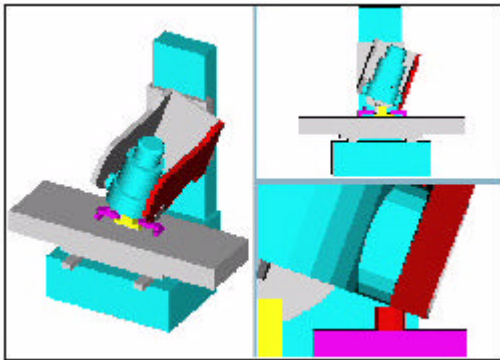
Travel Limits tab features

Axis Priority tab features

Machine Settings "Collision Detect" tab features

Features on this tab control when collisions between machine components are detected, which components are protected, and tolerances used for detecting collisions. (See how)

Colliding components are highlighted using the red **Error** color, and errors are issued to the Log file identifying collision causing block(s) and machine components.



Collision Detection On—When active, detects collisions between specified components.

Ignore Collision between Cutter and Stock—Controls when collisions between the cutter and the Stock component are ignored. This feature is useful when collision detection is desired between the stock and shank or holder portions of the tool assembly in the machine view, but not with the cutter. Options are:

- **No**—(default) Does not ignore cutter-stock collisions. All collisions are reported.
- **All Tools**—Ignores cutter-stock collisions for all tools, even inactive tools in multi-tool machines.
- **Active Tool**—Ignores cutter-stock collisions for the active tool. However, collisions between stock and inactive tools are detected.

Default Near Miss—Specifies the default collision tolerance applied to all collision cases when **Set All** is pressed (see below).

Set All—Sets the default collision tolerance for all collision cases to the **Default Near Miss** value. You can edit the supplied tolerance for individual cases.

Component/Component collision list—Lists the component-to-component collision cases that are checked when collision detection is turned on.

Component 1-2—These features are used to specify the component-to-component collision cases to check. The **Tolerance** value controls how close the components are permitted to be before reporting a collision.

Note: Do not configure for collision detection between components that move (slide or rotate) against each other, such as connected motion axes. In these cases, errors may occur each time the components move.

Tolerance—Specifies the tolerance to use for collision checking. Enter a positive value to be alerted if components come near each other within the specified clearance, zero to indicate components may not touch, or a negative value if components are expected to collide by the specified value.

Add—Adds a new collision case to the list.

Modify—Modifies an existing collision case in the list.

Delete—Deletes the selected collision case from the list.

Machine Settings "Tables" tab features

Machine Tables are used to specify important machine locations and detail tool change motion. For additional information about tables, see one of the following topics:

What do you want to do?

Learn about configuring tables

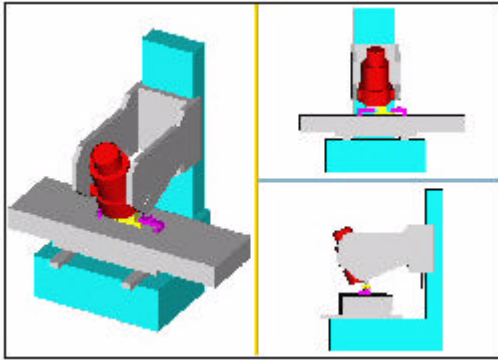
See how to add, modify, or delete tables

See reference information about tables, such as: list of tables, how they are used, data entry, etc.

Machine Settings "Travel Limits" tab features

Features on this tab define how far each machine axis can go, and control when travel limit errors are detected. (See how)

Overtravel errors cause the violating machine component to "light up" in the **Overtravel Color** and errors are output to the Log file identifying the problem component.



Overtravel Detection On—When active, turns on overtravel detection (travel limit checking). Components that move beyond specified limits are highlighted in the **Overtravel Color** and errors are issued to the Log file identifying error causing block(s) and machine components.

Overtravel Color—Color in which components that move past their specified limits are shaded. Available shade colors are defined on the **File menu > Colors: Define** tab.

Component Min/Max travel list—List of motion components in the machine and corresponding axis limits.

Minimum / Maximum—Specifies the minimum and maximum travel limits for the motion axis selected in the list

Ignore—When selected, ignores overtravel checking for the corresponding axis.

Modify—Modifies the selected component/travel entry in the list to have the values configured in the window.

Machine Settings "Axis Priority" tab features

Features on this tab control how machine axes move in rapid positioning mode (e.g. G0). By default, machine axes are interpolated with equal priority to arrive at their destinations at the same time. You can change the priority to simulate non-interpolated rapid motion such as "dog leg" or "squared off" movement. (See how)

Single—This option provides a single set of rapid priorities are used. The rapid priorities applied to the tool moving in the negative Z direction are the same as those applied to the tool moving in the positive Z direction.

Pos/Neg—This option provides different rapid priorities This option provides depending on the tool moving in a positive or negative Z direction. With this option, two rapid priority lists are provided. Use the **Positive / Negative** options to control which list is seen.

Positive / Negative options—When **Pos/Neg** priority is selected, these options control which rapid priority list is seen: the list applied when the tool is moving in the **Positive** Z direction, or the list for **Negative** Z tool motion.

Axis Priority list—List of all machine motion axes and corresponding priorities.

Priority—Priority for moving motion axes move in rapid. A value of "1" moves that axis first. A

Help on VERICUT

"2" moves the axis with secondary priority (after those with priority "1"), and so on. Axes having the same priority move together.

Interpolated—When active, interpolates axes that move together such that they arrive at their destination at the same time. Clear this checkbox to have axes move based on their defined **Rapid Rate**- ref. **Model menu > Model Definition: Component Attributes** tab.

Analysis menu functions

X-Caliper

Location: **Analysis menu > X-Caliper**

Toolbar shortcut: 

Opens a window that displays measurements, historical, and mathematical information about the VERICUT model. X-Caliper shows what is being measured by highlighting the "measure from" point (+) and the "measure to" point (o). Results from X-Caliper operations are displayed in the X-Caliper window. Measurements are described with respect to the active tool path coordinate system.

Note: Machined features can only be measured in a workpiece view.

X-Caliper window features:

Feature/History tab—Displays feature (mathematical description) and history related to machine cuts, including: tool path file name, record number, and tool path record text responsible for the cut. This function is useful to spot-check machined features, and determine the source of an error cut identified in red by VERICUT. (See how)

Stock Thickness tab—Features used to measure solid stock material. (See how) XYZ values specify the point to measure from. IJK values control the measurement direction into material.

Air Distance tab—Features used to measure air gap between two solid surfaces. (See how) XYZ values specify the point to measure from. IJK values control the measurement direction away from material.

Closest Point tab—Features used to measure the shortest distance from an XYZ point location to the closest point on the model surface. (See how)

Volume tab—Calculates values for the cut workpiece volume (cubic units), original stock model volume (cubic units), and percentage of material removed from the original stock model. (See how)

Distance Angle tab—Features used to measure the distance and/or angle between any combination of 2 points, planes, or conic axes. (See how) See window features.

Scallop tab—Features used to measure the height of a scallop formed by parallel intersecting cylinders, typically cut by cutter corner radius. (See how) See window features.

Compute—Computes measurements based on the current X-Caliper window configuration. This feature is useful to have X-Caliper determine measurements based on features defined via keyed-in information.

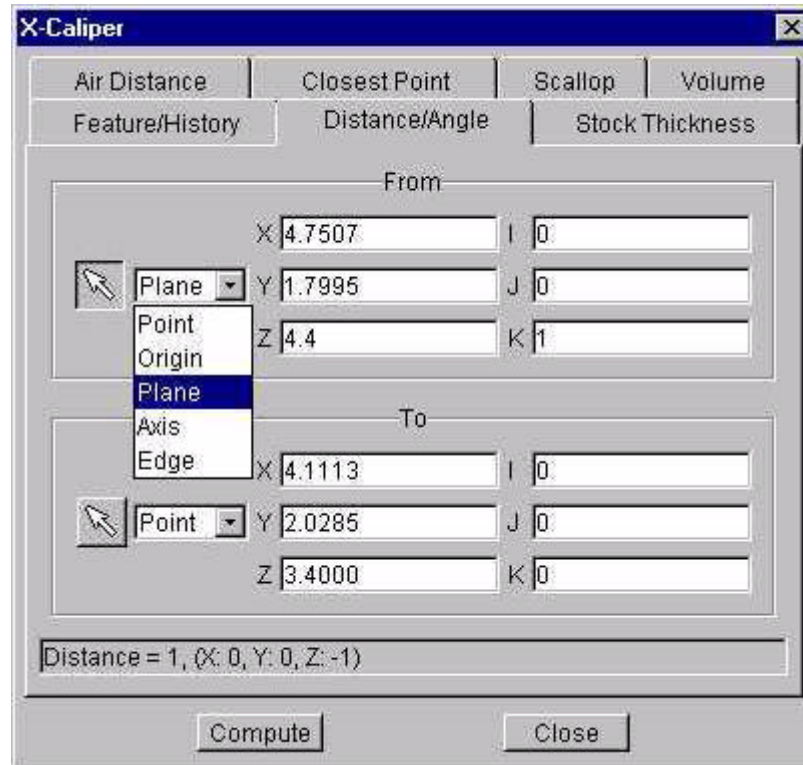
X-Caliper "Distance/Angle" tab features

The features on this tab are used to measure the distance and/or angle between two selected features. In general, select the "From" feature, then select the "To" feature-VERICUT automatically displays the

measurement data in the X-Caliper window. (See how)

Feature option lists—Controls types of features to measure between. Use the choice arrow to control which fields are filled in by screen pick data.

Sample X-Caliper window with "From" features option list exposed:

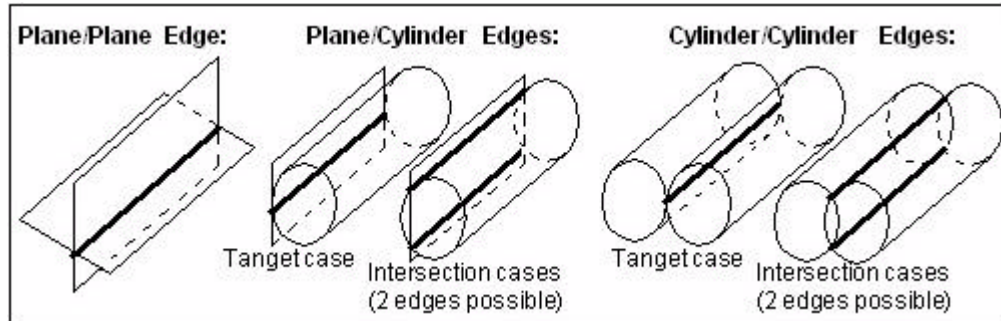


Features available to measure between:

- **Point**—XYZ point location.
- **Origin**—Origin of a selected model. To see the model origin, use the **View menu > Axes** function.
- **Plane**—A flat, unbounded surface. If a curved surface is selected, data retrieved represents a plane tangent to the selected model surface at the pick point.
- **Axis**—Centerline of a cylinder or conic.
- **Edge**—Edge formed by two intersecting planes or parallel cylinders. Special cases follow.

Plane/Cylinder edge- Only a cylinder with an axis parallel to the plane can produce an edge. Tangent conditions produce 1 edge. Intersecting conditions produce 2 possible edges, thus the one closest to the cylinder pick-point is used.

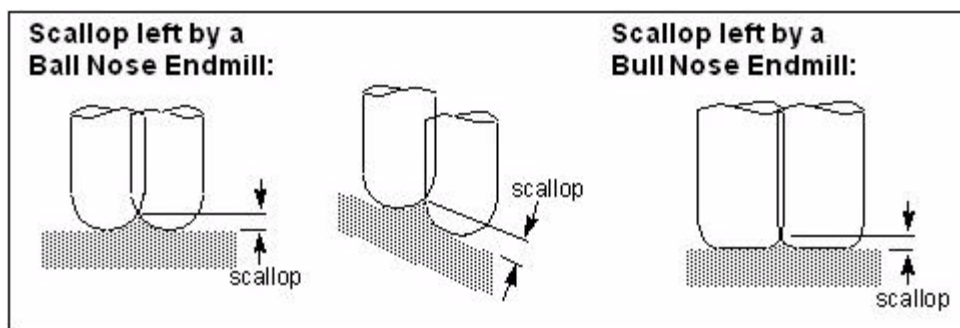
Cylinder/Cylinder edge- Only cylinders with parallel axes can produce an edge. Tangent conditions produce 1 edge. Intersecting conditions produce 2 edges, thus the one closest to the last cylinder pick-point is used.



XYZ, IJK data fields—Data from the screen pick fills the fields in the corresponding "From" or "To" area, depending on the active choice arrow. If necessary, you can edit data in these fields, then press **Compute** to obtain the desired measurement.

X-Caliper "Scallop" tab features

The features on this tab are used to measure the height of a scallop formed by two parallel intersecting cylinders. (See how)



Cylinder 1 / Cylinder 2—Cylinders to measure between. Use the choice arrow to control which fields are filled in by screen pick data.

Center / Axis / Radius—Uneditable text fields that display feature data related to cylinders selected for scallop measurement.

Scallop Height—Uneditable text field where the scallop measurement is displayed. This field is automatically updated after picking a feature for **Cylinder 2**.

AUTO-DIFF

Location: **Analysis menu** > **AUTO-DIFF**

Toolbar shortcut: 

Opens a window to compare design model data against the VERICUT simulated cut model. Using AUTO-DIFF clearly identifies discrepancies between the cut model the intended design model. Errors are identified before actual machine time is wasted machining an incorrect part. This function can also be used to protect previously cut surfaces when new stock is added.

Three types of design models can be compared: solid models, surface data, and inspection points. The **Comparison Method** controls which type of design data is compared, while **Comparison Type** determines the error condition checked for. Features on the **Settings** tab control the inspection tolerances and colors used to shade errors detected by the AUTO-DIFF operation. AUTO-DIFF operations are performed interactively, in batch, or automatically via a special comment records in the tool path file (ref. "PPRINT/VERICUT-COMMAND record" in the Supported APT-CLS records section of the *VERICUT Files* chapter).

Note: The degree to which VERICUT can detect discrepancies is affected by the **Cutting Tolerance** (ref. **File menu** > **Properties: Tolerance** tab). For best results set this tolerance to be smaller than the tolerance(s) to be used for AUTO-DIFF comparisons.

Tip: Use other VERICUT functions, such as **Analysis menu** > **X-Caliper** and **Zoom** (ref. **View menu** > **Orient**) to help determine the source of errors detected by AUTO-DIFF.

AUTO-DIFF window features:

Settings tab features

Tolerance tab features

Boundary tab features

Cut Options tab features

Compare—Compares the design model(s) specified by the AUTO-DIFF window settings.

Restore—Restores the VERICUT model as it was before performing the AUTO-DIFF operation.

Report—Opens a window containing an AUTO-DIFF Report file (inspection report) identifying errors detected by the AUTO-DIFF operation. Information in this window includes a listing of tool path record numbers responsible for the errors, and amount of deviation from the design model. Selecting an error in the list highlights the location of the error on the model.

AUTO-DIFF "Settings" tab features

Stock Display / Design Display—When active, displays the stock model (or design model) in the AUTO-DIFF view. Options are provided to display these models as a solid or translucent.

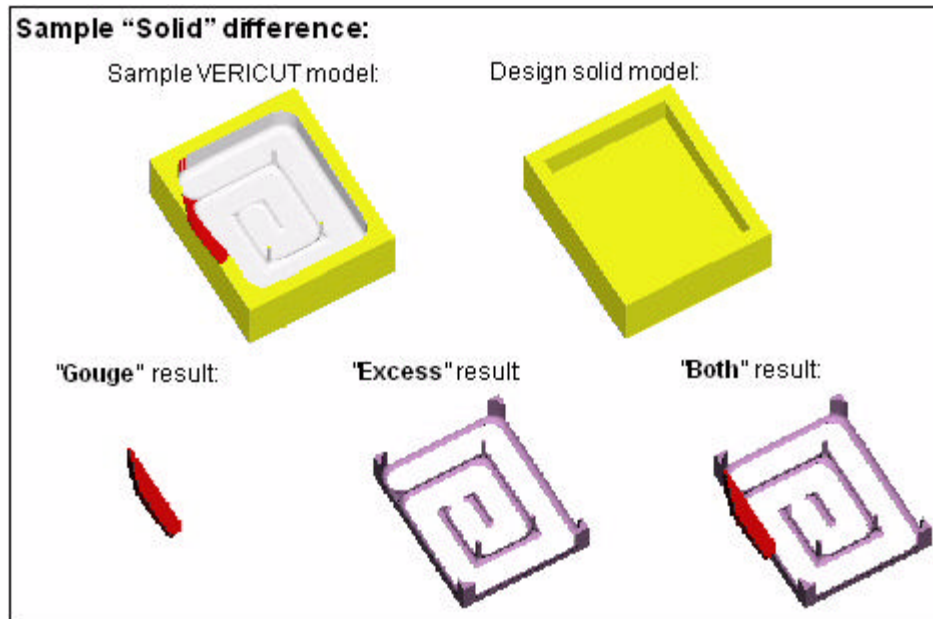
Comparison Method—Controls the method of comparing design model data against the

VERICUT model. The comparison method also reconfigures the Settings tab to reflect the tolerances and other data required to compare the data as requested. All methods generate an AUTO-DIFF model that can be analyzed or saved as an In Process file.

Regardless of the method chosen, the design model must first be loaded prior to comparison (ref. **Model menu > Model Definition: Model** tab). Design models can be loaded at any time.

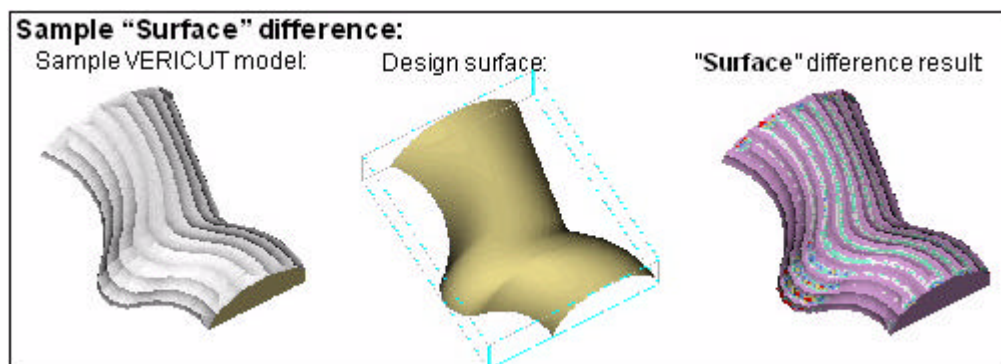
AUTO-DIFF methods:

- **Solid**—Compare a solid design model by subtracting solid models from each other. Examples follow.



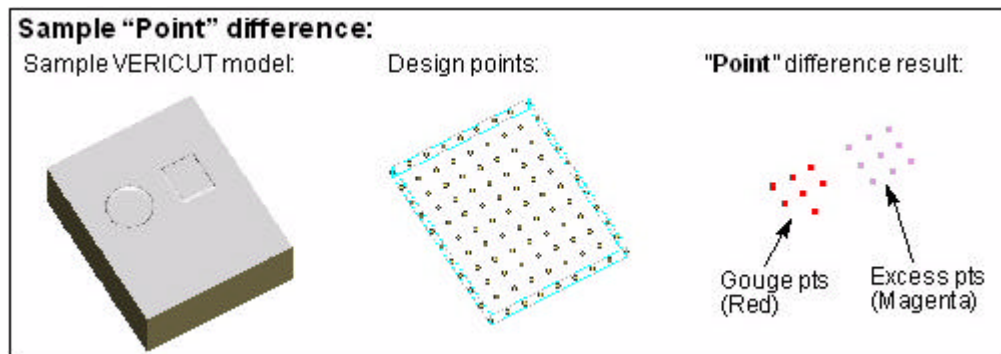
Hint: Check solid design model integrity by loading it as a stock model, then analyze the **Volume** of the stock model. The following problems indicate the design model need repair, or is suitable only for the **Surface** comparison method: holes appear in the VERICUT model, the model partially displays, or does not display at all.

- **Surface**—Compare a surface, set of surfaces, or skin of a solid design by overlaying the surface on the cut model. VERICUT only compares where the design surface and cut model overlap. Surface comparison provides multiple colors and tolerances that produce a "rainbow" affect to show the extent of errors. An examples follows.



- **Point**—Compare a set of design inspection points by overlaying the points on the cut model.

VERICUT searches from each point for an intersection with the model surface. Surface normal vectors control the direction in which to search. Only points found to be related to errors are displayed.



Comparison Type—Controls the error type(s) checked for by the AUTO-DIFF operation. Options are: **Gouge**, **Excess**, or **Both**.

AUTO-DIFF "Tolerance" tab features

Depending on the active **Comparison Method**, the features and actions in this window are different.

Solid comparison tolerances:

Gouge—Tolerance used when checking for gouges during a **Solid** comparison. Gouges detected are shaded in the associated gouge color.

Excess—Similar to **Gouge** above, except applies when checking for excess material.

Surface comparison tolerances:

Gouge Range/Color list—Range of tolerances used when checking for gouges during a **Surface** comparison. You may have up to seven tolerances listed in decreasing order from top to bottom. Gouges detected are shaded in the gouge color(s) associated with the depth of the gouge.

Excess Range/Color list—Similar to **Gouge** above, except applies when checking for excess material.



Point comparison tolerances:

Gouge Tolerance—Tolerance used when checking for gouges during a **Point** comparison. Gouges detected are shaded in the gouge color specified for a **Solid** comparison.

Excess Tolerance—Similar to **Gouge** above, except applies when checking for excess material conditions.

Gouge Check Distance—Maximum distance to check for the cut model surface in the gouge direction. Distance is measured from the inspection point along direction of the normal vector used with the inspection point.

Excess Check Distance—Similar to **Gouge Check Distance** above, except applies when checking for excess material.

Default Normal—Default normal vector in which to check during a **Point** comparison when vector data is not present in the Design Points file. Enter three values separated by spaces, for example "0 0 1".

Normal Direction—Indicates if surface normal vectors associated with design points are pointing outward (**Out**) or inward (**In**) relative to the design model.

Check Limits—When active, limits points compared to those located within the specified boundaries. Design points located outside of the boundaries are not checked. Clear this checkbox to compare all design points.

Minimum XYZ / Maximum XYZ—Specifies the limits of the boundaries defining the zone where design points are compared. Specify boundary locations using three values relative to the world coordinate system, separated by spaces.

AUTO-DIFF "Boundary" tab features

Features on this tab are used to generate and export geometry representing boundaries of excess material detected by a **Surface** method comparison on a model cut in the FastMill cut mode (ref. **File menu** > **Properties**).

Output File—Name of the file to receive exported boundary data.

File Type—Type of output file to create. Options are:

- **IGES**—IGES profile data. The way in which boundary curves are represented are controlled via the active **Curve Type** (see below).
- **Points**—Output points in GOTO/ X, Y, Z format representing boundary corner points. Comments in the output file identify separations between closed boundary curves. When exported points are imported into a CAD system, excess material boundaries can be recreated via drawing lines from point to point in their listed order.

Curve Type—Controls the IGES geometry type used to represent boundary curves. The curve type choices are: **NURBS** and **Polyline**.

Model Coordinates—Coordinate system (**World** or **Toolpath**) to export boundary data relative to.

Units—Unit measurement system (**Inch** or **Millimeter**) for exported boundary data.

Minimum Gap—Smallest allowable gap between boundaries. If independent boundaries are

closer to each other than the specified distance they are joined.

Minimum Area—Smallest excess material area that is bounded. Smaller excess material areas are not bounded. Zero bounds all excess material areas.

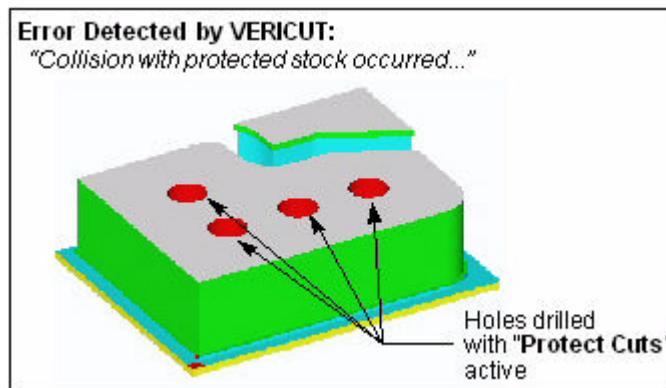
Project to Plane—When active, projects boundaries in the Z direction to the top plane of the stock model. Clear this checkbox to leave the boundaries in 3-D space where they were created.

Output—Outputs the boundary curve data according to the current window settings.

AUTO-DIFF "Cut Options" tab features

Protect Cuts—When active, protects previously cut surfaces from additional machining. Protection is applied after a subsequent Stock model has been loaded.

Zero gouge tolerance is used to check for gouges in previously cut material. VERICUT notifies you immediately about a gouge in previously cut surfaces via the red **Error** color and an error is issued to the Log file.



OptiPath menu functions

OptiPath Control

Location: **OptiPath menu > Control**

Toolbar shortcut: 

Opens a window to control when tool paths are optimized by OptiPath, specify the OptiPath Library file containing the desired OptiPath records, and choose the stock material and NC machine involved in the cutting the workpiece.

Feed rates and spindle speeds can be monitored via the **Info menu > Status** window. A summary of the cutting conditions encountered during optimization and min/max optimized feed rates is written to the Log and Results files.

OptiPath Control window features:

On / Off / Interactive—Controls when tool paths are optimized. Options are:

- **On**—Turns tool path optimization on and writes optimized tool path data to the **Optimized File**. For optimization to occur, OptiPath records must be associated with cutting tools in the tool path file. There are 2 methods of doing this:

Method 1- associate the OptiPath tool description and number of teeth with a tool in a Tool Library file, then choose the stock material and machine in the OptiPath Control window.

Method 2- choose the stock material and machine in the OptiPath Control window, then build a tool list that associates OptiPath records with cutting tools to be optimized in the tool path file.

With either method, tools without cutting tool-to-OptiPath record associations are not optimized. If both methods are used, associations in the tool list override those stored with tools in the Tool Library.

- **Off**—Turns optimization off. Optimization is automatically turned off when the end of the tool path file has been reached, or when the tool path has been reset or rewound.
- **Interactive**—Turns tool path optimization on and interactively prompts for optimization data at each tool change. As you enter data for optimizing the cuts performed by each tool, VERICUT automatically stores this information in a OptiPath Library file and sets tool properties to point to the new OptiPath records.

OptiPath Library—Name of the OptiPath Library file containing the desired OptiPath records.

Optimized File—Specifies the name of the modified (optimized) tool path file to receive optimized tool path data. The format of this file, APT vs. G-code, is the same as the tool path file being optimized. When an APT tool path type is optimized the programmed “FEDRAT” records are commented out using “\$\$” and written to the modified path for reference. The **Optimized File** name must be different from that of the original tool path file. OptiPath will not permit the original tool path file to be overwritten.

When optimizing multiple tool path files an optimized tool path file is output for each tool path file simulated. Each output file has an ascending sequence number appended to the end of the base file name. For example: assume **Optimized File**=“part.mco” is entered (without the

quotes). Optimizing 2 tool paths outputs the following files:

part001.mco => optimized tool path from tool path 1

part002.mco => optimized tool path from tool path 2

If a dot “.” is the last character of the optimized tool path file name, then three digits are appended to the file name as the extension. Continuing with the above example: assume **Optimized File**=“part.” is entered. Optimizing the same 2 tool paths outputs the following files:

part.001 => optimized tool path from tool path 1

part.002 => optimized tool path from tool path 2

If an asterisk “*” is used in the optimized tool path file name, the original tool path file name is used as the output file name in place of the asterisk. If characters follow the asterisk, the input tool path file name extension is replaced with the characters following the asterisk. Examples follow.

Original tool path file:	Optimized File name:	Output file name:
dir1/filename.tp	dir2/*	dir2/filename.tp
dir1/filename.tp	dir2/*.op	dir2/filename.op
dir1/filename.tp	dir2/*op	dir2/filename.op
dir1/filename.tp	dir2/*op.tp	dir2/filename.op.tp
filename.tp	*.op	filename.op

Only a single asterisk is acted on in the optimized tool path file name, with one exception: asterisk dot asterisk (“*.”). This case behaves exactly like the single asterisk file name described above. This is NOT a regular expression replacement, therefore any characters before the asterisk are ignored and lost. When there is more than one asterisk, only the first one is replaced. Examples follow.

Original tool path file:	Optimized File name:	Output file name:
dir1/filename.tp	dir2/*.*	dir2/filename.tp
dir1/filename.tp	dir2/op*	dir2/filename.tp
dir1/filename.tp	dir2/op*.ext	dir2/filename.ext
dir1/filename.tp	dir2/*op.*	dir2/filename.op.*

Material—Identifies the stock material being cut. The material name is changed via clicking the arrow and selecting from a list of available materials, or clicking in the data field and typing a new material name.

Machine—Identifies the NC machine being used to cut the workpiece. This feature functions similar to **Material** described above

Note: For optimization to occur the **Material** and **Machine** names must match those of the desired records in the OptiPath Library.

Override All Feedrates—Overrides optimized feed rates calculated for all tools optimized. 100%

Help on VERICUT

uses optimized feed rates as is, 50% cuts feed rates in half, 200%” doubles the feed rates, and so on.

OptiPath Manager

Location: **OptiPath menu > Manager**

Toolbar shortcut: 

Opens a window to create and maintain OptiPath Library files containing OptiPath records that describe how to optimize feed rates and/or spindle speeds for various tools and cutting conditions. The OptiPath Library data is used by VERICUT during tool path optimization.

The OptiPath Library summary list shows the available OptiPath records. The OptiPath Manager shows the optimization values for the selected tool in the right side of the window. The values can be used as is, or change values and press **Apply** to update the record in the library.

Each record is identified by a unique combination of the following data (identifiers):

- Stock material to be cut
- Machine that will be used to cut the workpiece
- Description of the cutting tool
- Number of cutting teeth on the tool

The Tool Manager shows the selected tool in the Tool Display area.

For optimization to occur OptiPath records must be associated with cutting tools in the tool path file. There are 2 methods of doing this:

Method 1- associate the OptiPath tool description and number of teeth with a tool in a Tool Library file, then choose the stock material and machine in the OptiPath Control window.

Method 2- choose the stock material and machine in the OptiPath Control window, then build a tool list that associates OptiPath records with cutting tools to be optimized in the tool path file.

With either method, tools without cutting tool-to-OptiPath record associations are not optimized. If both methods are used, associations in the tool list override those stored with tools in the Tool Library.

OptiPath Manager window features:

File menu—Functions in this menu create a new OptiPath Library, open or save OptiPath Libraries, and exit the OptiPath Manager.

Material/Machine/Tool Description/# Teeth lists—The four pieces of data that uniquely identify an OptiPath record in the library. Data for each identifier can be entered, or in the case of **Material** and **Machine**-can be selected from option lists.

Add—Adds a new record to the OptiPath Library.

Copy—Copies an existing record in the OptiPath Library to create a new record.

Delete—Deletes the selected tool from the OptiPath Library.

Specify Temporary Cutter Shape—Opens a window to describe the shape of the cutter to optimize. The cutter shape automatically appears when a tool in the Tool Library is defined with

Help on VERICUT

OptiPath properties that link to the selected OptiPath record. VERICUT refers to the cutter shape to calculate material removal volume and other machining-related data via features located on the **Feed/Speed** tab.

Features in this window are similar to those found in the Tool Manager - see "Tool icons and descriptions" under **Setup menu > Tool Manager** in this chapter for more information.

Unit Power—Power required to remove one cubic unit of material per minute. The power specification differs, depending on the units set for the OptiPath record (ref. **Settings** tab below).

Accel/Decel—When active and a G-code tool path is being optimized, this feature applies acceleration/deceleration settings from the NC machine configuration to adjust optimized feed rates. VERICUT looks ahead in the tool path file and slows the machine prior to direction changes, as well as changes feed rates based on the machine axes ability to accelerate and decelerate.

Options are:

- **Off**—Does not adjust optimized feed rates for acceleration/deceleration.
- **Exact Stop**—Assumes all axes must stop and be positioned exactly at the programmed location before the next block is processed. This option is recommended when sharp corners are required between motion blocks.
- **Continuous**—Allows small errors in positioning in order to maintain a more constant velocity during corners or direction changes.

Feed/Speed tab features

Settings tab features

Spindle tab features

Plunge tab features

Entry/Exit tab features

Angle tab features

Depth Table tab features

Width Table tab features

Notes:

- The block preceding an un-optimized block is not processed through accel/decel logic.
- Accel/decel settings apply to all linear and circular motions, including those in air.

OptiPath Settings "Feed/Speed" tab features

Cutting Condition:

This group of features are used to describe various machining conditions to optimize. As values are changed, the cutter graphic is updated to reflect the current cutting condition. The unit of measure for

values entered are determined by the active units on the **Settings** tab.

Axial Depth / Radial Width—Depth and width of cut.

Feed per Minute—Feed rate calculated from feed per tooth, spindle speed, and number of cutting teeth on the cutter.

Feed per Tooth—Thickness of material (aka. "chip load") removed by each tooth on the cutter.

Optimization Method:

Chip Thickness—Optimizes cutting by flat, ball, and bull nose endmills based on maintaining a specified chip thickness. This method increases the feed rate when width of cut is less than 50% cutter diameter, or when depth of cut is less than the tool corner radius. Enter the desired chip thickness in the data field right of the option. This option is recommended for optimizing semi-finishing and finishing operations with carbide cutting tools, and high speed machining. It is especially effective for mold and die applications machining hard materials.

Note that chip thickness equals feed per tooth when cutting with 50% or more of the cutter diameter, and cutting deeper than the tool corner radius. If the chip thickness is not known, unselect all optimization methods and use the **Cutting Condition** sliders to determine this value.

Volume Removal—Optimizes based on a constant volume of material to be removed. Enter the constant volume removal rate which is to govern feed rate optimization. Volume removal rate is expressed in cubic inches or millimeters, depending on the active units for the OptiPath record. If the volume removal rate is not known, unselect all optimization methods and use the **Cutting Condition** sliders to determine this value.

Spindle Speed—Spindle speed specified in revolutions per minute.

Surface Speed—Optimizes feed rates and spindle speeds based on constant surface speed (CSS). Spindle speeds are varied to maintain a constant surface speed at the maximum tool contact diameter, and feed rates maintain a constant feed per tooth load in material. Enter desired CSS value in the data field right of the option. Use surface feet/minute (FM) for inch tool paths, surface meters/minute (MM) for metric tool paths. If the CSS value is not known, unselect all optimization methods and use the **Cutting Condition** sliders to determine this value. Use features on the **Spindle** tab to limit RPM fluctuations and specify the feed per tooth load to be maintained.

This option is recommended for optimizing finish machining by ball or bull-nosed endmills, and high speed machining. CSS optimized tool paths are recommended only for use on high speed NC machines capable of varying spindle speed quickly without harming spindle motor hardware.

Spindle Power—Power required to remove material. The value entered differs depending on the active units for the OptiPath record:

- Inch units= horsepower required to remove 1 cubic inch of material
- Metric units= kilowatts required to remove 1 cubic centimeter of material per second

Air Cut Feed Rate—Optimizes based on reducing the amount of time the tool spends moving in air. This method applies the **Air Cut Feedrate** when the tool is not in contact with material and uses the programmed feed rate for all tool motions that are in contact with material. The feed rate entered should reflect the maximum speed at which all machine axes can move simultaneously in feed mode (e.g. G01).

- **Default**—When selected, sets the default **Air Cut Feed Rate** as follows:

G-code tool path = **Max Feed Velocity** of the machine's X-axis.

APT tool path = **Fast Feed** value.

Optimize by Tables—Optimizes based on specified feed rates for various cut depths, widths, and angles. Enter values into the tables on the **Depth Table**, **Width Table**, and **Angle** tabs to control optimization by this method.

Fill—Automatically fills table data fields with values calculated using the features on this tab. The following values are supplied:

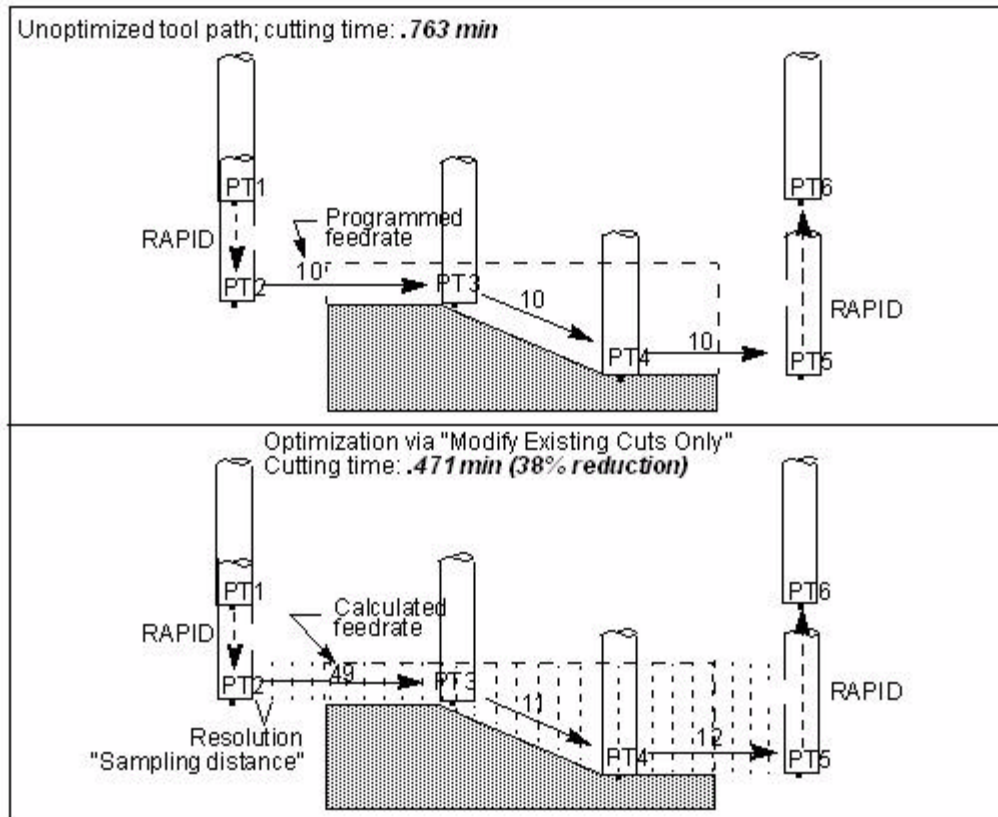
- **Depth Table** tab—defines five equally spaced cut depths based on cutter height and calculates a cutting feed rate for each depth based on volume of material removed.
- **Depth Table** tab—supplies the calculated volume removal rate for the value for **Max. Volume Removal Rate**.
- **Width Table** tab—calculates a percentage of calculated feed rate for each **% Width** value based on volume of material removed.

OptiPath Settings "Settings" tab features

Modify Existing Cuts Only—When selected, this option calculates the most efficient feed rate for each motion. Using the **Resolution** value on the **Settings** tab each motion is divided into samples, then each sample is analyzed and an optimum feed rate is calculated. The feed rate calculated for the sample with the most severe cutting conditions is retained as the optimized feed rate for that motion.

This option is recommended when disk space is limited for storing optimized tool path data, or when tool positions are relatively close together. A tool path optimized with this choice has more efficient feed rates and the same or slightly more records than the original.

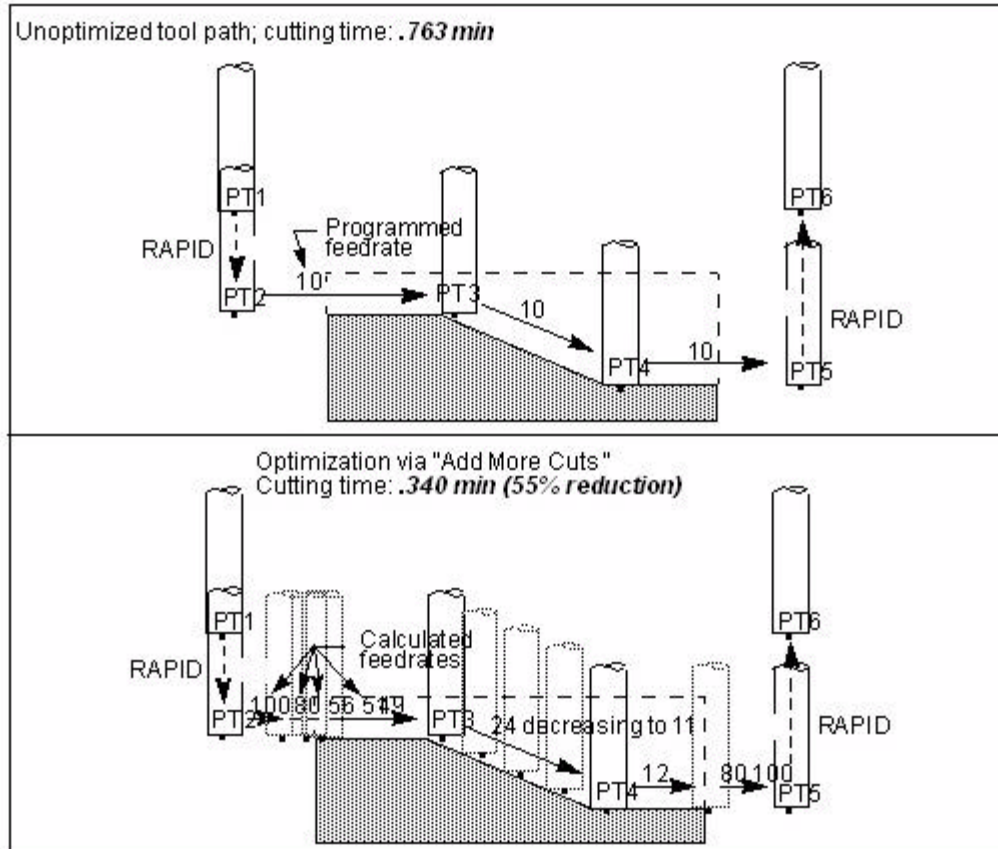
Sample optimization results using the **Modify Existing Cuts Only** option:



Add More Cuts—Similar to **Modify Existing Cuts Only** except motions are divided up when cutting conditions warrant feed rate changes. This option is good when disk space is ample, or when optimizing widely varying cutting conditions, such as: intermittent cutting, deep versus shallow cuts, etc.

A tool path optimized with this choice has very efficient feed rates, but many more records than the original. The **Minimum Feedrate Change** value controls the quantity of feed rates output to the optimized tool path file.

Sample optimization results using the **Add More Cuts** option:

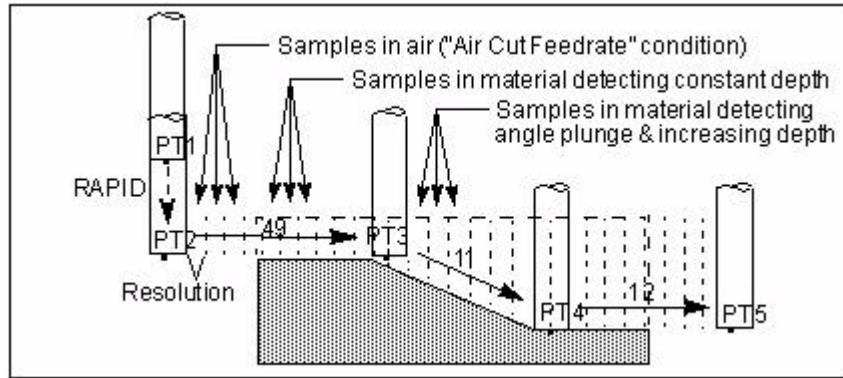


Inch / Millimeter option—Controls the unit measurement system (**Inch** or **Metric**) in which optimization data is entered. If the OptiPath record units are different from the session units, the record values are converted appropriately when loaded.

Resolution—Controls the frequency or “sampling distance” used to analyze each tool path motion. Each feed rate controlled motion is partitioned into samples based on this distance. The samples are then analyzed during feed rate calculation. Options are:

- **Distance**—Uses the sampling distance in the corresponding data field. Distance is measured along the tool motion. Using a resolution equal to 1/2 to 1/4 of the tool diameter is recommended.
- **Auto**—Automatically calculates the resolution distance based on the size of the cutting tool relative to the displayed model. This choice is recommended when little or no knowledge of cutting methods or feed rates is available.

Example of **Resolution** applied when optimizing tool path motions:



Minimum Feedrate Change—Specifies the minimum change from the current optimized feed rate that will cause a new optimized feed rate to be output. This feature controls the quantity of feed rates output to the optimized tool path file. A small value causes more optimized feed rates to be output than when a larger value is entered.

- **Default**—When selected, sets the default **Minimum Feedrate Change** = 1 IPM or 25 MPPM, whichever applies.

Clean-up Feedrate—Feed rate used when the tool is adjacent to, but not removing material. This condition is commonly referred to as a “spring pass”. **Tip:** Set the **Clean-up Feedrate** to be less than the **Air Cut Feedrate** since actual cutting leaves small amounts of material from tool deflection.

- **Default**—When selected, sets the default **Clean-up Feedrate** as follows:

G-code tool path = 50% of the **Max Feed Velocity** of the machine's X-axis.

APT tool path = 50% of the **Fast Feed** value.

Minimum Cut Feedrate—Specifies the minimum optimized feed rates that can be output when removing material.

- **Default**—When selected, sets the default **Minimum Cut Feedrate** = 1 IPM or 25 MPPM, whichever applies.

Maximum Cut Feedrate—Specifies the maximum optimized feed rates that can be output when removing material.

- **Default**—When selected, sets the default **Maximum Cut Feedrate** as follows:

G-code tool path = 45% of the **Max Feed Velocity** of the machine's X-axis.

APT tool path = 45% of the **Fast Feed** value.

Circle Feedrate—Controls when and how feed rates are optimized for circular motions (e.g. G02-3). Circle feed rate optimization produces a single optimized feed rate for the entire circular motion regardless of the active **Modify Existing Cuts Only** or **Add More Cuts** option.

Options are:

- **Programmed**—Uses the original programmed feed rate.
- **Fixed**—Uses the feed rate in the corresponding data field.
- **Optimize**—Calculates an optimized feed rate based on the maximum material removed

throughout the circular motion.

Circle feed rates are not optimized under the conditions listed below—these circles receive the original programmed feed rate.

- helical (spiral) circle motions
- circle motion where the tool axis is not perpendicular to the circle axis
- G-code data circle motion programmed in inverse time feed rate mode (e.g. G93)

NURBS Feedrate—Controls when and how to optimize NURBS motion feed rates (e.g. G06, G06.2). The active **Modify Existing Cuts Only** or **Add More Cuts** option does not affect feed rates calculated by this feature.

Options are:

- **Programmed**—Uses the original programmed feed rate.
- **One Feedrate**—Calculates the most efficient feed rate for NURBS motions. The feed rate calculated for the most severe cutting condition used as the optimized feed rate.
- **Multi Feedrates**—Calculates multiple feed rates for NURBS motions when cutting conditions warrant feed rate changes.

Feedrate Wear Reduction—Reduces the calculated feed rate based on the amount of material that has been removed. The value entered is the percentage of reduction to be applied after each cubic unit of material removed. Zero indicates no adjustment for wear.

Example: assume "**Feedrate Wear Reduction=.2**". The calculated feed rate is reduced .2% after each cubic unit of material has been removed. After five cubic units removed the calculated feed rate is reduced 1% (.2% x 5).

OptiPath Settings "Spindle" tab features

The features on this tab limit RPM fluctuations and specify feed per tooth load maintained during **Surface Speed** optimization.

Minimum RPM Change—Specifies the minimum change from the current optimized spindle speed required to output a new spindle speed. This feature controls the quantity of spindle speed changes output to the optimized tool path file. Small **Minimum RPM Change** values cause more spindle speed changes to be output than larger values.

- **Default**—When selected, sets the default **Minimum RPM Change** = 1000.

Tip: Start with a value such that only 3-4 changes can occur across the RPM range.

Minimum RPM—Specifies the minimum spindle speed that can be output during optimization.

- **Default**—When selected, sets the default **Minimum RPM** as follows:

G-code tool path = machine spindle minimum set by *SpindleSpeedMin* macro, if not set then 100 RPM.

APT tool path = 100 RPM.

Maximum RPM—Specifies the maximum spindle speed that can be output during optimization.

- **Default**—When selected, sets the default **Maximum RPM** as follows:

G-code tool path = machine spindle maximum set by *SpindleSpeedMax* macro, if not set then 10000 RPM.

APT tool path = 10000 RPM.

OptiPath Settings "Plunge" tab features

The features on this tab control the optimized feed rate for motions along the tool axis that plunge into or retract from material.

Plunge Feedrate—Controls when and how feed rates are calculated for plunging along the tool axis into material. Options are:

- **Off**—Disable plunge feed rate control.

- **Feed/Minute**—Uses the feed rate entered in the corresponding data field.

- **% of Prog.**—Adjusts programmed feed rates based on the percentage entered in the corresponding data field. 100% uses the programmed feed rate as is, 50% cuts the feed rate in half, etc.

Note: Plunge Feedrate overrides **Angle** table feed rate for 90 degree cutting angle (plunge motion), or the **Entry Feedrate** for entry on plunge motions.

Clearance Distance—Specifies the distance from material to instate the feed rate for plunging.

Retract Feedrate—Controls when and how the feed rates are calculated for retracting along the tool axis away from material. Options are:

- **Off**—Disable retract feed rate control.

- **Feed/Minute**—Uses the feed rate entered in the corresponding data field.

- **% of Prog.**—Adjusts the programmed feed rate based on the percentage entered in the corresponding data field. 100% uses the programmed feed rate as is, 50% cuts feed rate in half, etc.

Note: Retract Feedrate overrides **Angle** table feedrate for -90 degree cutting angle (retract motion).

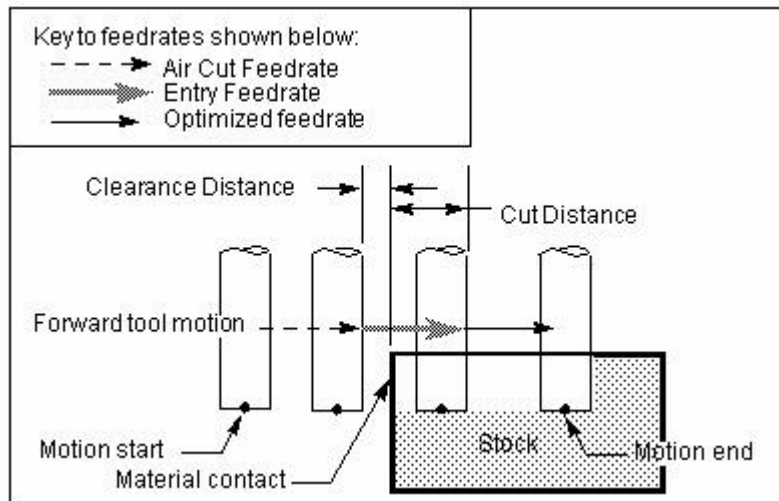
OptiPath Settings "Entry/Exit" tab features

The features on this tab control the optimized feed rate for entering and exiting material. The feed rate takes affect at a specified distance from material contact and remains in affect until the tool has cut a specified distance into material, or the entry motion ends.

Entry Feedrate—Controls when and how feed rates are calculated for entering material.

- **Feed/Minute**—Uses the feed rate entered in the corresponding data field.
- **% of Prog.**—Adjusts the programmed feed rate based on the percentage entered in the corresponding data field. 100% uses the programmed feed rate as is, 50% cuts feed rate in half, etc.
- **% of Calc.**—Similar to **% of Prog.**, except adjusts the calculated feed rate.

Affect of **Entry Feedrate** applied on a motion entering material:



Notes:

- When entry motions are optimized with the **Modify Existing Cuts Only** choice, calculated entry feed rates are considered among other cutting conditions to determine the appropriate feed rate for motions which enter material.
- **Plunge Feedrate** overrides entry controlled feed rate for entry on plunge motion.

Clearance Distance—Specifies the distance from material to instate the feed rate for entering.

Cut Distance—Specifies the distance to cut into material using the **Entry Feedrate**, after which normal feed rate optimization resumes.

Exit Feedrate—Controls when and how the feed rates are calculated for exiting material. Options are same as those described above for **Plunge Feedrate**.

Cut Distance—Specifies the distance away from breaking out of material to use the **Exit Feedrate**.

Clearance Distance—Specifies the distance after breaking free from material to continue using the feed rate for exiting, after which the **Air Cut Feedrate** resumes.

OptiPath Settings "Angle" tab features

The features on this tab adjust the optimized feed rate for 3-axis linear cuts ramping in material. While

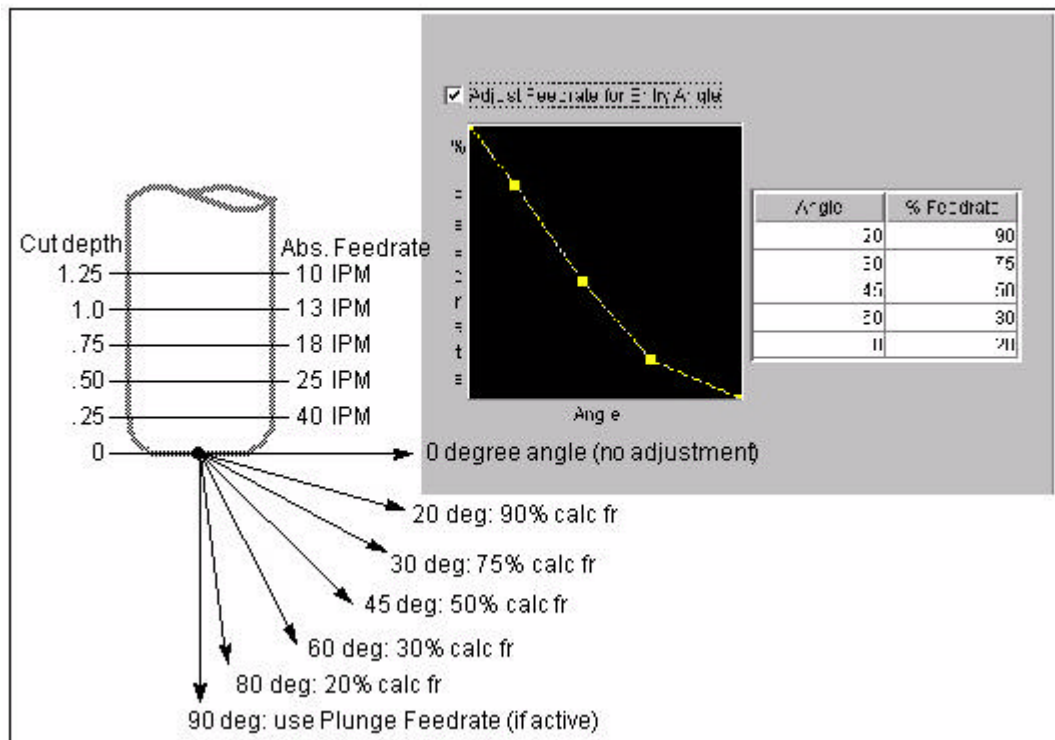
these features are normally used to reduce the feed rate for ramping downward, they can also be used to affect feed rates for upward ramping motions.

Adjust Feedrate for Entry Angle—When active, adjusts feed rates for ramping motions based on the table configuration.

Angle / % Feedrate table—The **Angle** column lists up to five cut angles in descending order to describe the range of expected cut angles. Angle values can be entered, or supplied via dragging the control points on the graph. Values can be from -90 to 90, where “0” represents horizontal milling, “90” represents plunge motion along the tool axis, and “-90” represents retract motion. Add after or delete angle fields via right-clicking on them and choosing the appropriate option.

The **% Feedrate** column lists the adjustments to apply to calculated feed rates when ramping in material at the corresponding angle. 100% uses the calculated feed rate as is, 50% cuts feed rate in half, etc. Cuts at angles less than the minimum listed in the table receive the adjustment specified for the minimum angle. Similarly, when the maximum specified cut angle is exceeded, the adjustment specified for the maximum angle is used.

A sample **Angle** table follows.



Notes:

Angle adjusted feedrate for 90 degree cutting angle is overridden by **Plunge Feedrate** if active.

Planar milling motions (0 degree angle) are not adjusted.

OptiPath Settings "Depth Table" tab features

The features on this tab controls how optimized feed rates are calculated for cutting at various depths during **Optimize by Tables** optimization. The calculated feed rate can be used as is, or adjusted using the **Angle** and/or **Width Table**, as well as features on other tabs in this window.

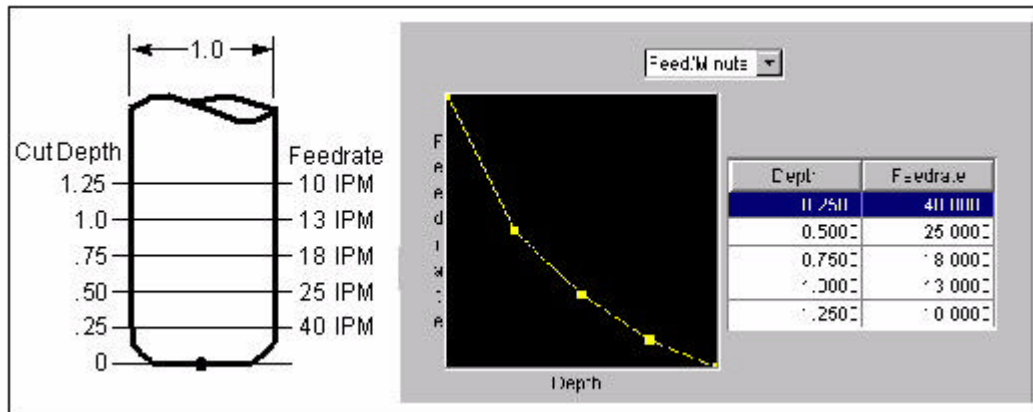
Depth Feedrate value type option list—Controls how **Depth** table feed rate values are interpreted. Options are:

- **Feed/Minute**—Calculates optimized feed rates based on absolute feed rates specified for various cut depths. Specify feed in units specified on the **Settings** tab.
- **% of Prog.**—Calculates optimized feed rates based on percentages of programmed feed rates specified for various cut depths. “100” uses the programmed feedrate as is, “50” cuts the feed rate in half, etc.
- **Feed/Tooth**—Calculates optimized feed rates based on feed per tooth, or chip load specified for various cut depths. Specify feed per tooth in units used by the tool path file (IPR or MMRP).

Depth / Feedrate table—The **Depth** column lists up to five cut depths in descending order to describe the range of cutting by the tool. Depth values can be entered, or supplied via dragging the control points on the graph. Add after or delete depth fields via right-clicking on them and choosing the appropriate option.

The **Feedrate** column lists the feed rates to apply when the tool is cutting at the corresponding depths. Feed rates are interpreted differently based on the active depth feed rate value type (see the option list above). Cuts at depths less than the minimum listed in the table receive the feed rate specified for the minimum depth. Similarly, when the maximum specified cut depth is exceeded, the feed rate specified for the maximum depth is used and an error identifying the condition is written to the Log file.

A sample **Depth Table** follows.



Tip: Unselect all optimization methods and use the **Cutting Condition** sliders to determine appropriate feeds and speeds for a known successful cutting condition, then select **Optimize by Tables** and press **Fill** to automatically enter **Depth Table** values.

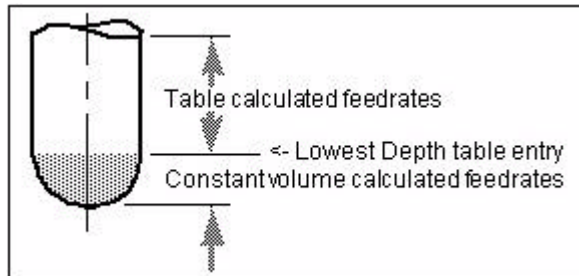
Feedrate Control Options—Controls when and how volume removal rate is considered in feed rate calculations during optimization by tables.

Options are:

- **Off**—Disable volume removal rate considerations.
- **Const. Volume Below Min. Depth**—Specifies a constant volume of material is to be removed

when the depth of cut is less than the lowest **Depth Table** entry. The volume removal rate is expressed in cubic units, where “units” refers to the units specified on the **Settings** tab. If the volume removal rate is not known, unselect all optimization methods and use the **Cutting Condition** sliders to determine this value.

The example below shows the affect of **Constant Volume Below Minimum Depth** on feed rate calculations for a ball-nose endmill when the lowest **Depth Table** entry is equal to the radius of the tool.



- **Max. Volume Removal Rate**—Specifies the maximum volume removal for the tool. Optimized feed rates are bounded such that the rate at which material is removed never exceeds this value. The volume removal rate is expressed in cubic units, where “units” refers to the units specified on the **Settings** tab. If the volume removal rate is not known, unselect all optimization methods and use the **Cutting Condition** sliders to determine this value.

OptiPath Settings "Width Table" tab features

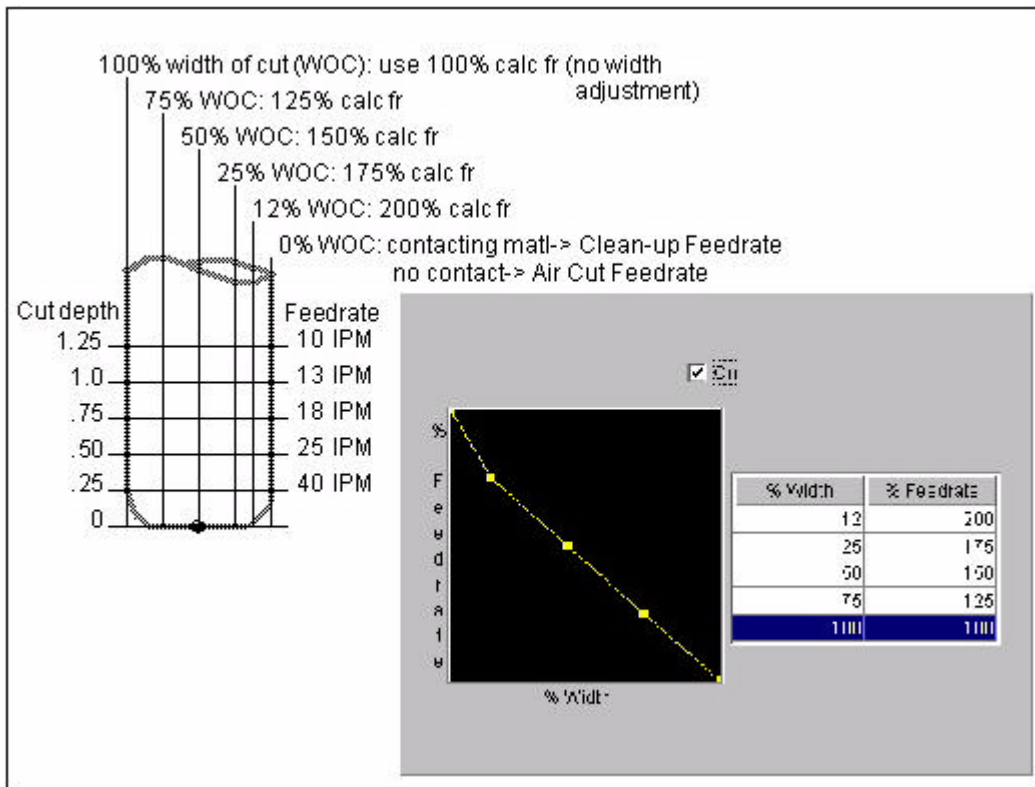
The features on this tab control the adjust feed rate for cuts of varying width. When the tool is cutting with less than full diameter the feed rate can typically be higher than for a full width cut, aka "slot cut".

On—When active, adjusts feed rates for cuts of varying width based on the table configuration.

% Width / % Feedrate table—The **% Width** column lists up to five cut widths in increasing width order to describe the range of expected cut widths. Width percentages (percent of cutter diameter) can be entered, or supplied via dragging the control points on the graph. Add after or delete width fields via right-clicking on them and choosing the appropriate option.

The **% Feedrate** column lists the adjustments to apply to calculated feed rates when removing material with the corresponding cut width. 100% uses the calculated feed rate as is, 50% cuts feed rate in half, etc. Cuts at widths less than the minimum listed in the table receive the adjustment specified for the minimum width. Similarly, when the maximum specified cut width is exceeded, the adjustment specified for the maximum width is used and an error identifying the condition is written to the Log file.

A sample **Width Table** follows.



DataLink

Location: **OptiPath menu > DataLink**

Opens a window that configures VERICUT to pass material removal data calculated by OptiPath to another application. It is up to the user site to have developed an application which can receive and utilize the material removal data.

Note: The OptiPath data link does not function while OptiPath is turned on (ref. **OptiPath menu > Control**).

OptiPath Data Link window features:

Command—Command used to execute the software application which will receive material removal data output through the OptiPath data link. The use of this field is optional. A blank field indicates that the software application has been previously executed and is awaiting material removal data to be output.

The default command “\${CGTECH_VCHOME}dlserver” refers to a “data link server” application provided by CGTech that directs material removal data to the screen. Viewing this data, users can see the output format and determine how data can best be used.

Host—Identifies the computer which is to run the software application intended to receive material removal data. The entry “localhost” specifies the software application resides on the same computer from which VERICUT has been executed.

Port—Specifies the computer port through which material removal data is to be passed to the software application.

Sample Distance—Distance along each motion at which material removal data is output. Each feed rate controlled motion is divided into samples based on this distance, then analyzed to calculate the material removal data.

Start—Starts (executes) the application specified in the **Command** field. When the field is blank VERICUT assumes the application has been previously executed and attempts to connect to it via the **Host** and **Port** information. In either case, if the start is not successful, an error message is issued to the OptiPath Data Link window and the start attempt is terminated.

Stop—Stops the software application that was started via pressing **Start**.

View—Similar to **Start**, except starts the application specified in the **Command** field such that it does not expect to be sent any data.

Tables Reference

Tables - listed alphabetically

The following VERICUT windows provide access to tables:

Setup menu > G-Code > Settings: Job Tables tab—Specify work offsets and machine locations specific to the current tool path file(s).

Setup menu > G-Code > Settings: Tool Tables tab—Store tool-related offset and compensation data.

Setup menu > Machine > Settings: Tables tab—Specify important machine locations and detail tool change motion.

For information about using Tables window features, see "About configuring tables" later in the *Using VERICUT* chapter.

The tables available in VERICUT are listed below in alphabetical order. See the appropriate section for details on data stored, data formats, and how to access each table.

Table name:	General description:
Base Work Offset table	Specifies the location from which work offsets are based
Cutter Compensation table	Stores cutter diameter compensation, or "CDC" offset values
Gage Offset table	Stores tool gage offset, or “gage length” values
Initial Machine Location table	Specifies where machine axes are initially located
Input Machine Zero table	No longer used
Input Program Zero table	Specifies the programmed zero location of a G-code tool path file
Machine Reference Location table	Specifies the reference location of machine axes
Part Transformation table	Specifies a workpiece transformation
RTCP Pivot Offset table	Specifies the offset between the Tool component origin and the tilting head pivot point
Tool Change Location table	Specifies the location of machine axes for changing the tool
Tool Change Retraction table	Specifies which machine axes retract for changing the tool
Tool Length Compensation table	Stores the tool length compensation offset values

Help on VERICUT

Work Offsets table	Stores work coordinate system offset (fixture offset) values

Base Work Offset table

This table specifies the location from which work offsets are based. Use this table when the tool path file references a base work coordinate system (e.g. G52) from which all other work offsets are based. (Ref. "Work Offsets table" in this section). Enter values to represent the work offset values entered at the NC control.

Data entry => 1: Xval Yval Zval Aval Bval Cval Uval Vval Wval

where linear offset values Xval-Zval and Uval-Wval are measured in the machine coordinate system from machine zero. Rotary offset values Aval-Cval are measured from the corresponding rotary component zero and are expressed in decimal degrees. Enter "0" for axes that do not have a base work offset or do not exist in the machine. Zero is assumed for unspecified machine axis offsets.

Examples that follow show table entry formats for various NC machine configurations.

Machine configuration	Table entry format
3 axes - XYZ	1: Xval Yval Zval 0 0 0 0 0 or 1: Xval Yval Zval ("0" assumed for axes A-W)
4 axes - XYZA	1: Xval Yval Zval Aval
4 axes - XYZB	1: Xval Yval Zval 0 Bval
4 axes - XYZC	1: Xval Yval Zval 0 0 Cval
4 axes - XZUW	1: Xval 0 Zval 0 0 0 Uval 0 Wval
5 axes - XYZAB	1: Xval Yval Zval Aval Bval
5 axes - XYZAC	1: Xval Yval Zval Aval 0 Cval
5 axes - XYZBC	1: Xval Yval Zval 0 Bval Cval
9 axes - XYZABCUVW	1: Xval Yval Zval Aval Bval Cval Uval Vval Wval

Cutter Compensation table

This table stores cutter diameter compensation, or "CDC" offset values used when simulating cutter diameter compensation. VERICUT access this table when the "**Process Cutter Comp.**" feature is active (ref. **Setup menu > G-Code > Settings: Settings** tab) and CDC data is processed, e.g. G41-42 Dn.

You can create a **Cutter Compensation** table via features on the **Setup menu > G-Code > Settings** window "**Tool Tables**" tab. Enter values to represent the CDC values entered at the NC control. Data entry formats follow.

Data entry => register_index: val

where "register_index" specifies the NC control register number where the corresponding CDC value offset "val" is stored.

Enter an offset value for each CDC offset register referenced by the tool path. Zero offset values are assumed when not specified, and do not require entry. The figure below shows a sample table to support a tool path file using CDC.

Sample NC data tool path and **Cutter Compensation** table entries

Sample NC data tool path & Cutter Compensation table:		
T1M6 G41X1Y2D1 : G40X0Y0 T2M6 G41X2Y3D2 : G40X0Y0 T3M6 G41X3Y4D3 : G40X0Y0 M30	NC control register data: D1: .5 D2: .375 D3: .25	Cutter Compensation table entries: 1: .5 2: .375 3: .25

How cutter compensation is simulated

Assume a control has been configured to support CDC, and a tool path file was programmed using "T1"-a zero diameter tool (sometimes known as "part edge programming"). Also assume a .5 diameter tool is used to cut the part. If the CDC offset programmed for use with this tool is "D1", the machine operator enters a value of .25 in the CDC register for "D1" to offset tool 1 when G41/42 is active. To simulate how this tool path will be compensated, a value of ".25" is entered into the **Cutter Compensation** table under **Index 1**, since this is the index referenced when "D1" is processed.

"Look ahead" logic, similar to Fanuc type "C" compensation, applies the CDC offsets during the simulation. The block activating CDC results in a "ramp-on" motion establishing the offset condition. The block cancelling CDC results in a "ramp-off" motion cancelling the offset condition.

Notes:

Output Precision specifies the accuracy for calculating CDC offset tool positions. (Ref. **Setup menu > Control > Settings > Motion**).

Tool paths can not be optimized by OptiPath when CDC offsets are being simulated.

Gage Offset table

This table stores tool gage offset, or “gage length” values. Use this table when the programming method of the tool path file drives a control point other than the tool tip, such as occurs with "Gage Length" and "Tool Length Compensation" programming methods.

You can create a **Gage Offset** table via features on the **Setup menu > G-Code > Settings** window "**Tool Tables**" tab. Data entry formats follow.

Milling data entry => register_index: 0 0 Zval

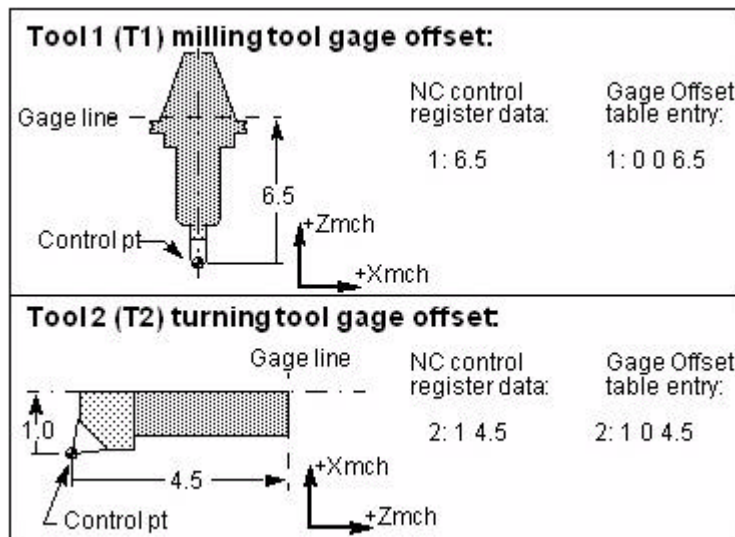
where “register_index” specifies the tool number and the gage length “Zval” is a positive value measured from the tool’s control point (typically the tool tip) to the gage line.

Turning data entry => register_index: Xval 0 Zval

where “register_index” specifies the tool number and the gage offset values Xval and Zval are measured from the tool’s control point to the tool spindle centerline and gage line, respectively.

Enter gage length values for each tool number referenced by the tool path file. The figure below shows sample table entries for a milling tool and a turning tool.

Sample **Gage Offset** table entries:



Note: **Gage Offset** table values override Gage Length or Gage Offsets tool properties stored in a Tool Library file.

Initial Machine Location table

This table specifies where machine axes are initially located. The machine is moved to this location when the machine is first loaded, when VERICUT is reset, or the tool path is rewound.

Data entry => 1: Xloc Yloc Zloc Aloc Bloc Cloc Uloc Vloc Wloc

where linear offset values Xval-Zval and Uval-Wval are measured in the machine coordinate system from machine zero. Rotary values Aval-Cval are measured from the corresponding rotary component zero and are expressed in decimal degrees. Enter "0" for axes which start at machine zero or do not exist in the machine. Zero is assumed for axes not having a specified initial location.

The following examples attempt to specify the initial location of a 5-axis XYZBC NC machine to be X=20 Y=62 Z=60 B=180 C=0:

1: 20 62 60 0 180 0 0 0 0

Correct- specifies X=20 Y=62 Z=60 A=0 B=180 C=0 U=0 V=0 W=0, even though AUVW are not in the machine configuration.

1: 20 62 60 0 180

Correct- specifies X=20 Y=62 Z=60 A=0 B=180; ACUVW are not specified and are assumed to be machine zero.

1: 20 62 60 180

Incorrect- specifies X=20 Y=62 Z=60 A=180; BCUVW are assumed to be at machine zero. B=180 has been incorrectly placed and is interpreted as A=180.

Input Machine Zero table

Table no longer used.

Input Program Zero table

This table specifies the programmed zero location of a G-code tool path file. This table is required when the machine operator establishes a zero location other than machine zero. This activity is also known as establishing a “floating zero” location.

Data entry => 1: Xloc Yloc Zloc Aloc Bloc Cloc Uloc Vloc Wloc

where linear axis locations Xloc-Zloc and Uloc-Wloc are measured in the machine coordinate system from machine zero. Rotary axis locations Aloc-Cloc are measured from the corresponding rotary component zero and are expressed in decimal degrees. Zero is used for axes which the tool path zero is the same as the machine zero, or do not exist in the machine configuration. Zero is assumed for axes not having a specified input program zero value.

Values for this table are calculated differently, based on the programming method used by the tool path file. Each programming method is listed below with an example of how **Input Program Zero** table values are calculated for that method.

With **Tool Tip** programming method- calculate values from the Machine reference point with no offsets in affect, such as: cutter compensation, work offsets, etc.

With **Gage Length** programming method- calculate values from the Machine reference point with no offsets in affect, such as: cutter compensation, work offsets, etc.

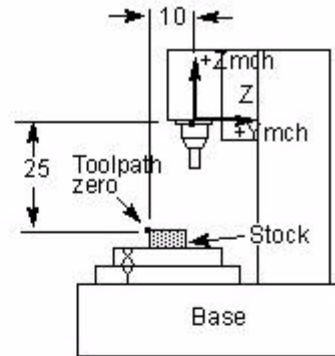
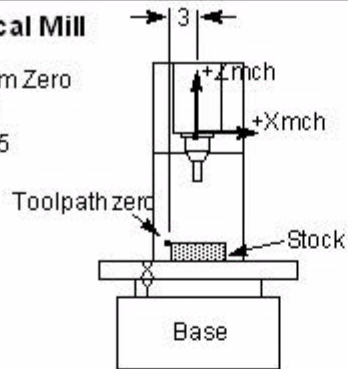
With **Tool Length Compensation** programming method- calculate values from the Machine reference point to the programmed zero when tool length compensation is active (e.g. G43).

The figure that follows shows sample table entries which specify the programmed zero location of tool path files run on various NC machine configurations.

3-axis Vertical Mill

Input Program Zero
table entries:

1: -3-10-25

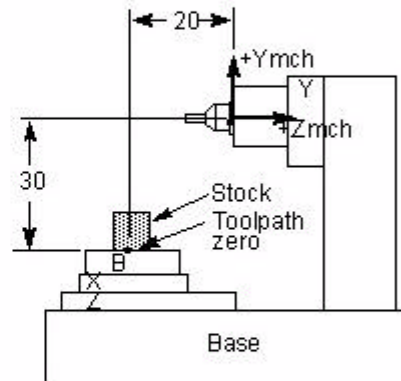
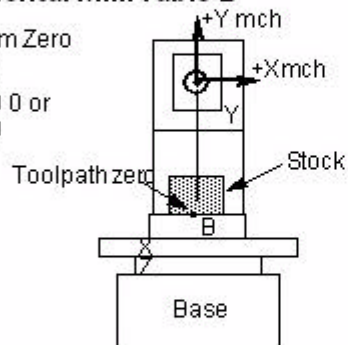


4-axis Horizontal Mill: Table B

Input Program Zero
table entries:

1: 0-30-20 0 or

1: 0-30-20

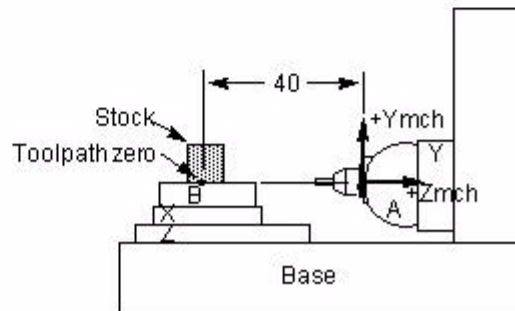


5-axis Horizontal Mill: Head A, Table

Input Program Zero
table entries:

1: 0 0-40 0 0 or

1: 0 0-40



Machine Reference Location table

This table specifies the reference location of machine axes. The reference location is where the machine axes move to when a “return to reference location” command is processed (e.g. G28), such as typically used with the **Tool Length Compensation** programming method.

Data entry => 1: Xloc Yloc Zloc Aloc Bloc Cloc Uloc Vloc Wloc

where linear axis locations Xloc-Zloc and Uloc-Wloc are measured in the machine coordinate system from machine zero. Rotary axis locations Aloc-Cloc are measured from the corresponding rotary component zero and are expressed in decimal degrees. Zero is used for axes having a reference location at machine zero or do not exist in the machine configuration. Zero is assumed for axes not having a specified reference location.

A return to reference point command typically cancels active offsets (tool length, work offsets, position set, etc.) and moves specified axes to their respective reference locations. The axis commands (e.g. X...Y...Z...A...B...C...) are processed prior to moving the machine to the reference location. For example, the command “G28 G91 X1 Y2 Z3” first moves the X axis by 1, Y axis by 2, and Z axes by 3, followed by moving to the X,Y,Z reference location.

Part Transformation table

This table specifies a workpiece transformation. Using this table and the *PartTransformation* macro, it is possible to automatically rotate, translate, and reverse the view of the VERICUT model during G-code processing.

Data entry => 1: rotX rotY rotZ posX posY posZ [1]

where rotX-rotZ rotate the VERICUT model, posX-posZ position the model. Including “1” in the seventh number location instructs VERICUT to reverse the model view.

To have VERICUT use the table information, use the “**Process Matrix**” on the **Setup menu > APT Settings** window: **Motion** tab.

RTCP Pivot Offset table

This table specifies the offset between the Tool component origin and the tilting head pivot point. Use this table when a rotary component located between the Tool and Base components has an Initial Machine Location table entry other than zero, and RTCP is to be used. This assumes that the rotary component initial location orients the tool axis such that no RTCP compensation is required.

Data entry => 1: Xval Yval Zval

where Xval-Zval are calculated by subtracting the location of the Tool component origin from the rotary pivot point location, with all axes at their initial machine location. Enter “0” for axes that do not have an offset or do not exist in the machine. Zero is assumed for unspecified offsets.

Ensure RTCP simulation is activated via the **Rotary Tool Control Point** feature in the **Setup menu > Control > Settings > Rotary** window.

Examples:

RTCP is desired with a vertical 5-axis tilting head machine that has a pivot length of 10 and a B-axis initial location of 180. This configuration puts the tool axis in a vertical orientation. For calculations, the location of the pivot point can be assumed to be at (0 0 0). This would require **RTCP Pivot Offset** table values: 0 0 10. $((0\ 0\ 0) - (0\ 0\ -10)) = (0\ 0\ 10)$

If the B-axis initial location was 0 (tool axis in a vertical orientation), then the **RTCP Pivot Offset** table is not needed. However, if the table was to be defined, it should contain the values: 0 0 10. $((0\ 0\ 0) - (0\ 0\ -10)) = (0\ 0\ 10)$

This table can also be used to support a machine/control where the post-processor takes care of the offset for the pivot length, and the control handles the offset for each tool's gage length. For this situation, set **RTCP Pivot Offset** table values to: 0 0 0.

Tool Change Location table

This table specifies the location of machine axes for changing the tool. This table is used when machine axes are specified to retract during a tool change via the **Tool Change Retract Method** feature (ref. **Setup menu > Control > Settings > Tooling**), or the Tool Change Retraction table.

Data entry => tool_index: Xloc Yloc Zloc Aloc Bloc Cloc Uloc Vloc Wloc

where the “tool_index” specifies the tool component number to which the tool change location values are to be used (typically “1” for single tool component machines). Linear axis locations Xloc-Zloc and Uloc-Wloc are measured in the machine coordinate system from machine zero. Rotary axis locations Aloc-Cloc are measured from the corresponding rotary component zero and are expressed in decimal degrees. Zero is used for axes which are located at machine zero, or do not exist in the machine configuration. Zero is assumed for axes not having a specified location.

The following examples specify the tool change location of a 5-axis XYZBC NC machine to be X=20 Y=62 Z=60:

1: 20 62 60 0 0 0 0 0 0 - specifies X=20 Y=62 Z=60 A=0 B=0 C=0 U=0 V=0 W=0, even though AUVW are not in the machine configuration

1: 20 62 60 - specifies X=20 Y=62 Z=60; ABCUVW are not specified and are assumed to be at machine zero

Tool Change Retraction table

This table specifies which machine axes retract for changing the tool. This table is used only with the **Use Retraction Table** option for the **Tool Change Retract Method** feature (ref. **Setup menu > Control > Settings > Tooling**). The retract location of each axes is specified via the Tool Change Location table.

Data entry => tool_index: Xval Yval Zval Aval Bval Cval Uval Vval Wval

where the “tool_index” specifies the tool component number for which the tool change retraction values are to be used (typically “1” for single tool component machines). For each machine axis (Xval-Wval), a value of 1 is entered when the axis is to be retracted, 0 is entered for no retract, or when the axis does not exist in the machine configuration. Zero is assumed for axes not having a specified value.

Examples:

1: 1 1 1 1 1 0 0 0 0 - specifies that the XYZAB axes are to be retracted to their tool change location while the CUVW axes (if they exist) are not to be retracted

1: 1 1 1 1 1 1 - as above, except CUVW are not specified and are assumed not to be retracted

1: 0 0 1 - specifies that only the Z axis is to be retracted

Tool Length Compensation table

This table stores the tool length compensation offset values (e.g. G43 **Hn**). Using this table is optional when simulating tool paths programmed in the "Tool Length Compensation" programming method. VERICUT automatically calculates tool length compensation values from the Input Program Zero and machine zero locations. Use this table to override values that are automatically calculated by VERICUT.

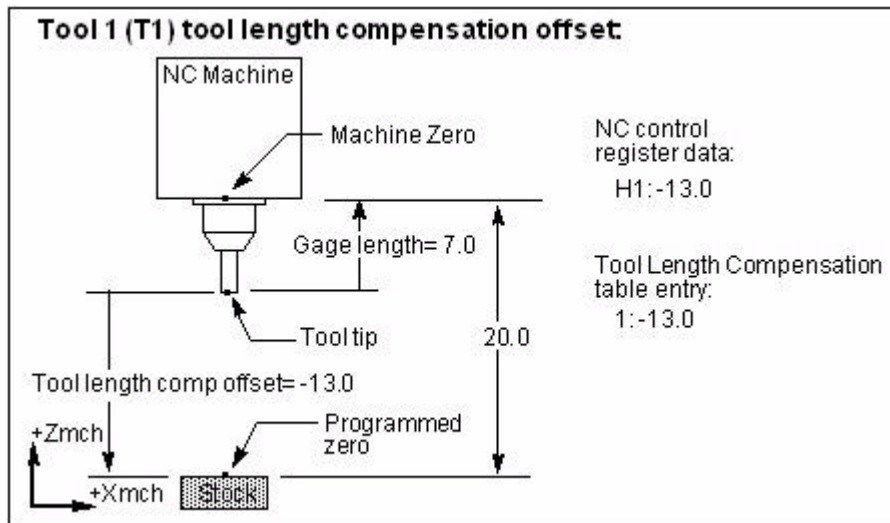
You can create a **Tool Length Compensation** table via features on the **Setup menu > G-Code > Settings** window "**Tool Tables**" tab. Enter values to represent the tool length compensation values entered at the NC control. Zero offset values need not be entered. Data entry formats follow.

Data entry format => register_index: val

where "register_index" specifies the NC control register number where the corresponding tool length compensation offset "val" is stored. The tool length compensation offset is usually a negative value measured from the control point (typically the tool tip) of each tool loaded at machine zero to the control point when the tool is positioned at the programmed zero.

Enter an offset value for each tool length offset referenced by the tool path file. The figure below shows a sample table to support a tool path using a tool with gage length=7.0 in a setup where the programmed zero is 20.0 from the machine zero.

Sample **Tool Length Compensation** table entry:



Work offset note: Work offsets have the affect of moving machine zero. When work offsets are in use, the tool length compensation offset value is calculated from the control point of each tool loaded at its respective work offset zero, to the control point when the tool is positioned at the tool path programmed zero.

Tool Nose Compensation table

This topic is still being written.

Turret Rotations table

This topic is still being written.

Work Offsets table

This table stores work coordinate system offset, or "fixture offset" values. Use this table when simulating programming with work offsets (e.g. G53-59). Enter values to represent the work offset values entered at the NC control. Work coordinate system offset values that are automatically loaded using via data in the tool path file (e.g. G10L2Pn) do not require entry.

Data entry => register_index: Xval Yval Zval Aval Bval Cval Uval Vval Wval

where "register_index" specifies the NC control register number where the corresponding work offset values "Xval Yval Zval..." are stored. Linear offset values Xval-Zval and Uval-Wval are measured in the machine coordinate system from machine zero. Rotary offset values Aval-Cval are measured from the corresponding rotary component zero and are expressed in decimal degrees. Enter "0" for axes that do not have a base work offset or do not exist in the machine. Zero is assumed for unspecified machine axis offsets.

Enter offset values for each local work coordinate system referenced by the tool path. The figure below shows a sample table to support a tool path file which references work offsets: G54, 55, 56.

Sample NC data tool path and Work Offsets table:

T1M6 G54X1Y2Z3 : : T2M6 G55X1Y2Z3 : : T4M6 G56X1Y2Z3 : G54X0Y0Z0 M30	NC control register data: 54: 0 0 0 55: 10 5 0 56: 20 10 0	Work Offsets table entries: 54: 0 0 0 55: 10 5 0 56: 20 10 0

The examples that follow assume a work offset register index of "54" and show general table entry formats for various NC machine configurations.

Machine configuration	Table entry format
3 axes - XYZ	54: Xval Yval Zval 0 0 0 0 0 0 or 54: Xval Yval Zval ("0" assumed for axes A-W)
4 axes - XYZA	54: Xval Yval Zval Aval
4 axes - XYZB	54: Xval Yval Zval 0 Bval
4 axes - XYZC	54: Xval Yval Zval 0 0 Cval
4 axes - XZUW	54: Xval 0 Zval 0 0 0 Uval 0 Wval
5 axes - XYZAB	54: Xval Yval Zval Aval Bval
5 axes - XYZAC	54: Xval Yval Zval Aval 0 Cval

Help on VERICUT

5 axes - XYZBC	54: Xval Yval Zval 0 Bval Cval
9 axes - XYZABCUVW	54: Xval Yval Zval Aval Bval Cval Uval Vval Wval

Notes:

To specify an initial work offset to be in affect, use **Setup menu > Control > Settings > Offsets**.

When a Base Work Offset table is in use, work offsets are established from the base work offset location (method commonly used by Phillips CNC controls).

Control Macros and Functions Reference

Macros - listed alphabetically

The macros which can be referenced via **Setup menu > Control > Word/Address** and the "Events" advanced control options are listed below in alphabetical order.

1. **3dToolOffset**—Renamed to: **Tool3dOffset**.
2. **3dToolXOffset, 3dToolYOffset, 3dToolZOffset**—Renamed to: **Tool3d[XYZ]Offset**.
3. **AAxisMachineMotion**—Sets the A value used to process the machine's A component. Similar to **AAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
4. **AAxisMotion**—Sets the A value used to command the machine's A rotary component.
5. **AbsoluteShift**—Shifts the work coordinate system by the specified absolute XYZ values. Any XYZ value which is not specified is assumed to be zero.
6. **AbsoluteShiftNum**—Similar to **AbsoluteShift**, except a value is passed which specifies which shift offset should be updated. (**AbsoluteShift** updates shift offset [0].) Three sets of shift offsets are supported: 0, 1, 2.
7. **ABType2CLS**—An Allen Bradley Type II subroutine call with no arguments. Same as **Type2CLS**, except it strips all characters starting with the "/" character. For example: (CLS,027sta/hd) would call subroutine file "027sta".
8. **ActivateSpindle**—Activates the Tool component with "Tool Index Number" equal to the value specified by the **SpindleValue** macro. Intended to replace the **ActiveTool1-5** macros.
9. **ActiveTool**—Activates the Tool component with "Tool Index Number" equal to the value specified by this macro. Intended to replace the **ActiveTool1-5** macros, and support using more than five tool positions on a machine.
10. **ActiveTool1, ActiveTool2, ActiveTool3, ActiveTool4, ActiveTool5**—Activates the Tool component with "Tool Index Number" of 1-5, respectively.
11. **AdditionalWorkCoord**—Updates the work coordinate system based on values retrieved from the Work Offsets table. The work coordinate index is used as an index into this table.
12. **ApplyPivotOffset**—Specifies whether the pivot offset should be applied. A value of 1=YES, 0=NO. When YES, the controller will compensate for the pivot distance when running in RTCP mode. The pivot distance can be specified with the RTCP Pivot Offset table. If the table is not specified, it will be calculated based on the location of the Tool and its corresponding Rotary component.
13. **ApplyTurretOffset**—Specifies whether the turret offset should be applied. A value of 1=YES, 0=NO. When YES, the controller will compensate for the turret offset. The turret offset is defined as the offset from the active tool to the zero point of the turret.
14. **AutoUpdateIPZ**—This macro is passed the text of the parent component name of the assembly that is being attached to or removed from the main machine assembly. Use the **Override Text** field to specify the component name. The offset will be calculated based on the difference between

the origins of the parent component (top of the component list) and the lowest level child component of the associated assembly.

The **Override Text** field is also used to pass a value of 1 or -1 to add or subtract the auto-calculated values from the internal Input Program Zero table offset. **Note:** This does not change any of the table values. The new value for IPZ can be seen using Modals/Debug/Debug Offsets.

Note: This macro should probably be executed after motion or on a block by itself. Also, since the internal value for Input Program Zero is being modified, this method will probably not work when running in Tool Length Comp mode.

15. **BaseWorkOffsetValues**—Sets the base work offset—an offset that is added to any current work offset. The XYZ offset is specified with the **WorkCoordXValue**, **WorkCoordYValue**, **WorkCoordZValue** macros.
16. **BAxisMachineMotion**—Sets the B value used to process the machine's B component. This macro is similar to **BAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
17. **BAxisMotion**—Sets the B value used to command the machine's B rotary component.
18. **BlockFinish**—Default event macro for the "End of Block Processing" event.
19. **BlockInit**—Default event macro for the "Start of Block Processing" event.
20. **BlockSkipSwitch1**—Specifies the value of the first Block Skip Switch variable. A value of 1=On, 0=Off.
21. **BoschType2CYCLE**—Sets the **ZAxisMotion**, the **CycleRapidLevelValue**, and the **CycleStepValue** based on the format of a Bosch Type II Cycle command.
22. **CallNCMacro**—Sets up the variables for calling a Fanuc style NC macro in the machine code input file. The machine code macro which is called is specified with the **SubroutineName** macro. This macro is also dependent on the calling of the **MacroVar** for setting the parameters which are to be passed to the subroutine. Up to 150 variable values can be passed by a call to an NC macro.
23. **CallNCMacroBlock**—Sets the mode to call a NC Macro on every block.
24. **CallNCMacroCancel**—Cancels the mode set with **CallNCMacroBlock** and **CallNCMacroMotion**.
25. **CallNCMacroMotion**—Sets the mode to call a NC Macro on every block with motion.
26. **CallSub**—Calls the machine code subroutine specified via the **SubroutineName** macro.
27. **CallSubName**—Calls the machine code subroutine specified with the input value.
28. **CallTextSubName**—Similar to **CallSubName**, except the subroutine name is interpreted as text, regardless of the **Type of Subroutine Names** control setting.

To use this macro, use the **Setup menu > Control > Word Format** function to define a "SubroutineName" word that takes an alpha-numeric argument. **Note:** It is critical that the value type be set to **Alpha-Numeric**. Then use **Setup menu > Control > Word/Address** to configure "SubroutineName" to call **CallTextSubName**.

29. **Cancel3dToolOffset**—Cancels the 3D Tool Correction offset (see **Tool3dOffset**).
30. **CancelAllWorkOffsets**—Cancels the shift offset (Fanuc G92), the work coordinate offset (G54-

59), and the Base offset (MAHO).

31. **CancelShiftOffsets**—Cancels the shift offset (Fanuc G92).
32. **CancelWorkOffsets**—Cancels the work coordinate system offset (G54-59)
33. **CAxisMachineMotion**—Sets the C value used to process the machine's C component. Similar to **CAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
34. **CAxisMotion**—Sets the C value used to command the machine's C rotary component.
35. **ChangeWorkCoord**—Defines the current position to be the specified XYZ value. This is an incremental shift from the current working coordinate system. Any XYZ value not specified is assumed to be zero.
36. **Cinci_5axisToolLengthComp**—Turns On/Off a special Cincinnati form of 5-axis tool length compensation. A value of 1=On, 0=Off.
37. **CinciRotateCenterCalc**—Calculate the center of rotation based off of the specified center and the mode set by **CinciRotateXYZMode**.
38. **CinciRotateXYZMode**—Cinci (Acramatic 950) specific macro that sets a parameter which indicates how rotation center positions should be interpreted.
39. **CircleCenterReset**—Resets the XYZ coordinates of the center of a circle. **Note:** The **CircleCenter[XYZ]** values are assumed to be modal. This macro resets these values.
40. **CircleCenterX, CircleCenterY, CircleCenterZ**—Sets the XYZ coordinates of the center of a circle, respectively. The Circles control settings determine how these values are interpreted.
41. **CirclePresentAxis**—When called with **Override Value** of “1”, supports processing circles using the linear axes present in the circle motion block. This macro should be called during the "Start of Processing" event, and is intended for use by controls on NC machines that have colinear axes and perform circular motions. Examples follow.

G2Y1.1242X-25.7855I.1171J.7743 - circle motion performed by Y & X linear axes

G2Y1.1242U-25.7855I.1171J.7743 - circle motion performed by Y & U linear axes
42. **CircleRadius**—Sets the radius of a circle. A negative value specifies that an arc of at least 180 degrees should be generated.
43. **CircleRadiusLarge**—Sets the radius of a circle such that an arc of at least 180 degrees will be generated. The input value is sign independent.
44. **ClampCompName**—Specifies the name of a clamp component (text value). This macro is typically used to clamp parts during pick-off spindle operations.
45. **ClampOnOff**—Specifies the state of the clamp specified with **ClampCompName**. A value of 0 is off, a value of 1 is on. This macro is typically used to clamp parts during pick-off spindle operations.
46. **CollisionOnCutPart**—Controls when collisions between machine components and the cut model are checked for. By default, collisions are detected when collision checking is turned on between the Stock and machine components. This collision checking can be turned off via calling **CollisionOnCutPart** with an **Override Value** of “0”, while “1” turns collision checking on again.

Note: “Near Miss” tolerances are not supported for collision checking against the cut model.

- 47. **ConnectCompName**—Sets the connect component name to the input text string value. Is used when dynamically connecting one component to another. Ref. **ConnectToCompName**.
- 48. **ConnectToCompName**—Connects the component specified with the **ConnectCompName** macro to the component specified by the input text string value.

Example: to have “M10” connect a component named “Stock” to a “U” axis component, use the **Setup menu > Control > Word/Address** function to define 2 groups as follows:

Word=M Range=10, Macroname=ConnectCompName, Override Text=Stock

Word=M Range=10, Macroname=ConnectToCompName, Override Text=U

- 49. **ConstantSurfaceSpeed**—Sets the Constant Surface Speed.
- 50. **ConstantSurfaceSpeedMode**—Sets the current mode to Constant Surface Speed.
- 51. **ConversionFinish**—Default event macro for the "End of Processing" event. Typically, added event macros should be placed before **ConversionFinish**.
- 52. **ConversionInit**—Default event macro for the "Start of Processing" event. Typically, added event macros should be placed after **ConversionInit**.
- 53. **CoolantFlood, CoolantMist**—Sets the coolant type to "Flood" or "Mist" and outputs "COOLNT/FLOOD" or "COOLNT/MIST", respectively.
- 54. **CoolantOff, CoolantOn**—Turns off/on the coolant and outputs the APT "COOLNT/OFF" or "COOLNT/ON" statement, respectively.
- 55. **CornerMode**—Defines the type of “auto-corner” to be generated, if any. Options are: 0 - for NONE, 1 - for CHAMFER, and 2 - for RADIUS.

The auto-corner capability uses “look ahead” logic for both CHAMFER and RADIUS. The resulting values are then passed on to cutter compensation logic. Corner RADIUS is only supported in the XY, YZ, or ZX planes. If the specified corner can not be achieved, a warning message is given and auto-cornering is ignored for that corner.

- 56. **CornerValue**—When in CHAMFER mode, this macro defines the length from the the programmed corner to the point at which the chamfer is to begin. In RADIUS mode, this defines the radius to be used in the corner. The start and end points of the arc will be calculated such that the arc is tangent to the cut coming into and out of the arc. In both cases, the original corner (as defined in the tool path) must be defined with two line segments.
- 57. **CutterComp3d**—Turns on 3d Cutter Compensation. See also: **Tool3dXOffset**.
- 58. **CutterCompDoubleLong**—Increases the movement amount by twice the tool offset value. The offset will be applied to each axis in which there is movement. (not modal)
- 59. **CutterCompDoubleShort**—Decreases the movement amount by twice the tool offset value. The offset will be applied to each axis in which there is movement. (not modal)
- 60. **CutterCompLeft**—Sets the cutter compensation to be Left.
- 61. **CutterCompLong**—Increases the movement amount by the tool offset value. The offset will be applied to each axis in which there is movement. (not modal)

- 62. **CutterCompOff**—Turns off the cutter compensation.
- 63. **CutterCompOffLookAheadX, CutterCompOffLookAheadY, CutterCompOffLookAheadZ**—Specifies where the next XYZ point is on the part (not the next GOTO point). These macros are called while ramping off of cutter compensation. These values are used to avoid gouging the part while ramping off. Look ahead cutter compensation will use these values to determine if the previous compensated point needs to be adjusted.
- 64. **CutterCompOffsetValue**—Sets the tool offset value based off of the corresponding entry in the Cutter Compensation table. (modal)
- 65. **CutterCompRight**—Sets the cutter compensation to be Right.
- 66. **CutterCompShort**—Decreases the movement amount by the tool offset value. The offset will be applied to each axis in which there is movement. (not modal)
- 67. **CutterCompToolNum**—Uses the current tool number to index into the Cutter Compensation table. The corresponding value is used as the compensation amount when CDC is turned on.
- 68. **CutterCompValue**—Uses the input value to index into the Cutter Compensation table. The corresponding value is used as the compensation amount when CDC is turned on.
- 69. **CutterCompValueDirect**—Sets the cutter compensation value to the input value. The corresponding value is used as the compensation amount when CDC is turned on.
- 70. **CutWire**—Cut/remove the EDM wire. This will output a “LOADTL/0” statement.
- 71. **CycleClearanceDistance**—Sets the cycle clearance distance. This overrides the default control setting.
- 72. **CycleDepth**—Sets the Cycle Depth value.
- 73. **CycleDepthInv**—If the Cycle Depth is to be interpreted as an absolute value, the Cycle Depth value is set to the input value. If the Cycle Depth is to be interpreted as an incremental value, the Cycle Depth value is set to the negated input value.
- 74. **CycleDepthType**—Specifies how the Cycle Depth is to be interpreted. 1= Absolute value, 2= Incremental value, 3= G-Code dependent.
- 75. **CycleDwellTime**—Sets the dwell time for a cycle. Dwell time is measure in revolutions or seconds (dependent on the feed mode).
- 76. **CycleIgnore**—Sets a flag which causes the cycle to not be executed on the current block.
- 77. **CycleIncDepthType**—Sets how an incremental Cycle Depth value is to be interpreted. 1= Relative to the Rapid Level, 2= Relative to the Initial Level, 3= Relative to the Part Surface.
- 78. **CycleIncRapidType**—Sets how an incremental Cycle Rapid value is to be interpreted. 1= Relative to the Rapid Level, 2= Relative to the Initial Level, 3= Relative to the Part Surface.
- 79. **CycleInitialOnMotion**—Specifies whether the cycle “Initial Point” should be initialized on each motion. 0= FALSE, 1= TRUE. (Default: FALSE).
- 80. **CycleRapidLevelInitial**—Similar to **CycleRapidLevelValue**, except that the value is specified by the current X, Y, or Z axis value, depending on the current plane.
- 81. **CycleRapidLevelInv**—Similar to **CycleRapidLevelValue**, except that the value is negated if the rapid value is to be interpreted as incremental.

- 82. **CycleRapidLevelValue**—Set the R point level.
- 83. **CycleRapidLevelValueMult**—Similar to **CycleRapidLevelValue**, except that the incoming value will be multiplied by the X/Y/Z multiplier, depending on the current G17,18,19 state.
- 84. **CycleRapidLevelZeroTracking**—Similar to **CycleRapidLevelValue**, except that it applies to zero-tracking machines.
- 85. **CycleRapidType**—Specifies how the Cycle Rapid value is to be interpreted. 1= Absolute value, 2= Incremental value, 3= G-Code dependent..
- 86. **CycleRetractInitial**—Sets the retract level position based on the current X, Y, or Z position (depending on the current plane).
- 87. **CycleRetractSpecifiedpoint**—Sets the retract level position to be the specified value, independent of the current plane.
- 88. **CycleRetractSpecifiedZT**—Similar to **CycleRetractSpecifiedpoint**, except that it applies to zero-tracking machines.
- 89. **CycleRetraction**—Set the Cycle Retract value. This is typically used when the retract level is an incremental distance from the clearance plane.
- 90. **CycleStepUpValue**—Sets the Cycle Step Up value. Use in cycle "Deep" mode.
- 91. **CycleStepValue**—Sets the Cycle Step Down value. Use in cycle "Deep" mode.
- 92. **CycleSubroutineOff, CycleSubroutineOn**—Turns on and off the calling of the specified subroutine on every motion, respectively. The name of the subroutine may either be numeric or text, based on the **Type of Subroutine Names** control setting.
- 93. **CycleSubroutineOnNoMotion**—Same as **CycleSubroutineOn**, except doesn't automatically turn motion on for the current block.
- 94. **CycleXyzDepth**—Specifies whether or not X,Y, or Z is used to specify cycle depth. 0= NO, 1= YES.
- 95. **CyclesBore**—Sets the cycle type to Bore.
- 96. **CyclesBoreDrag**—Sets the cycle type to Bore Drag.
- 97. **CyclesBoreOrient**—Sets the cycle type to Bore Orient.
- 98. **CyclesCancel**—Cancels the cycle mode.
- 99. **CyclesDeep**—Sets the cycle type to Deep.
- 100. **CyclesDrill**—Sets the cycle type to Drill.
- 101. **CyclesExecute**—Executes a single cycle (MAHO). All cycle parameters must have been previously set.
- 102. **CyclesFace**—Sets the cycle type to Face.
- 103. **CyclesMill**—Sets the cycle type to Mill.
- 104. **CyclesTap**—Sets the cycle type to Tap.
- 105. **CyclesThru**—Sets the cycle type to Thru.

- 106.**Debug**—Renamed to **DebugMacro**.
- 107.**DebugMacro**—Debugging macro that prints the corresponding word, the string associated with the value, and the value to the DEBUG destination.
- 108.**DefaultWire**—Sets the current tool number to be the default wire EDM tool number.
- 109.**DisableWorkShiftOffsets**—Disables the current Work and Shift offsets. The current values are saved, and can be enabled via a call to **EnableWorkShiftOffsets**.
- 110.**DoLoop**—Sets the input file locations of the do loop. During the first pass, the locations are saved. During the second pass, the locations are restored.
- 111.**Dwell**—Renamed to **DwellMacro**.
- 112.**DwellMacro**—Causes a dwell to occur. Typically called by G04, this macro must be called after the **DwellTime** macro. Dwell length is determined by the **DwellTime** variable value and the current feed mode (IPM/MMPM, IPR/MMPR):
- 113.**DwellRevolutions**—Similar to **DwellMacro**, except the dwell time value is always interpreted as revolutions.
- 114.**DwellSeconds**—Similar to **DwellMacro**, except the dwell time value is always interpreted as seconds.
- 115.**DwellTime**—Sets the dwell time variable. Typically called by the "X" word when a G04 is in the block, the dwell time variable is interpreted differently depending upon the macro called after **DwellTime**:
- **DwellMacro** -> seconds in IPM/MMPM mode, revolutions in IPR/MMPR mode
 - **DwellRevolutions** -> always revolutions
 - **DwellSeconds** -> always seconds
- 116.**DynamicWorkOffsets**—When called with an **Override Value** of "1", this macro rotates the work and shift offsets based on the rotation that is applied to the stock. An assumption is made that the entire amount of the dynamic and work offsets should be rotated by the amount the part is rotated. This logic will not work when the machine is not built about zero, and only part of the work offset is to be rotated.
- 117.**EDMCalcUpperGuideHeight**—Causes the upper wire guide height to be recalculated based off of changes to the **Lower2Reference** height, **Reference2Upper** height, or the distance between the guides.
- 118.**EDMCalcUpperGuideLocation**—Causes the position for the upper wire guide to be recalculated based off of changes to the **Lower2Reference** or **Reference2Upper** heights. This will actually cause a change in the position of the upper guide. Typically this macro would not be used in Wire EDM.
- 119.**EDMGuide2Guide**—Adjusts the Z component and upper wire guide height when the Z value specifies the distance from the lower to upper guides.
- 120.**EDMLower2Reference**—Specifies the distance from the wire EDM table top to the XY plane.
- 121.**EDMLowerGuideHeight**—Specifies the distance from the lower wire guide to the top of the table.

- 122.**EDMReference2Upper**—Specifies the distance from the wire EDM XY plane to the UV plane.
- 123.**EDMUpperGuideHeight**—Specifies the distance from the UV plane to the upper wire guide.
- 124.**EnableWorkShiftOffsets**—Enables the previously disabled work shift offset. See **DisableWorkShiftOffsets**.
- 125.**EndLoop**—Sets the input file locations of the end of a do loop. During the first pass, the locations are saved. During the second pass, the locations are restored.
- 126.**EndProgram**—Ends the processing of the current job. Outputs "FINI".
- 127.**EndProgramRewind**—Outputs "REWIND" and call **EndProgram**.
- 128.**EndSub**—Sets up the return values for a subroutine during pass1 or returns from the current subroutine during pass 2.
- 129.**ErrorMacro**—Creates an error message using the input text string value.
- 130.**FadalLcode**—If less then 100, the value sets the loop count. If the value is an even multiple of 100, it defines the beginning of a subroutine. Otherwise it is a call to a subroutine, with the last 2 digits being the loop count.
- 131.**FadalWorkCoord**—For Fadal, E54 is the same as E1. Therefore if the value is greater then 53, 53 is subtracted from the value.
- 132.**FeedInvTime**—Sets the feedrate based on the input value (InverseTime). The Feedrate will be calculated based on the formula: $\text{Feedrate} = \text{Distance} * \text{InverseTime}$.
- 133.**FeedInvTimeMultiplier**—Specifies a multiplier that is to be applied to the value specified with the **FeedInvTime** macro. (Default=1.0)
- 134.**FeedInvTimeSec**—Similar to **FeedInvTime**, except that the units are in 1/seconds rather than 1/minutes.
- 135.**FeedModeMinute**—Sets the feed mode to Feed per Minute.
- 136.**FeedModeRevolution**—Sets the feed mode to Feed per Revolution.
- 137.**FeedModeTime**—Sets the feed mode to Inverse Time. The Feedrate will be calculated based on the formula: $\text{Feedrate} = \text{Distance} * \text{InverseTime}$.
- 138.**FeedPerMinute**—Sets the feedrate based on the input value. The units will be either IPM or MPPM.
- 139.**FeedPerRev**—Sets the feedrate based on the input value. The units will be either IPR or MMPR.
- 140.**FeedRate**—Sets the feedrate based on the input value. It will used the previously set feedrate mode to output the corresponding units.
- 141.**Fidia_ConvInit**—Initializes the Fidial specific rotation axis and center to 0.0.
- 142.**Fidia_Rotate**—Turns on the Fidial specific rotation using the values set with the **Fidia_Rotate[XYZ]Axis** and **Fidia_Rotate[XYZ]Point** commands.
- 143.**Fidia_RotateCancel**—Cancels the Fidial specific rotation..
- 144.**Fidia_RotateXAxis, Fidial_RotateYAxis, Fidial_RotateZAxis**—Sets the Fidial specific X,Y,Z rotation values respectively.

- 145. **Fidia_RotateXPoint, Fidia_RotateYPoint, Fidia_RotateZPoint**—Sets the Fidial specific point of rotation of the X,Y,Z coordinate respectively.
- 146. **GLSetCurrentZeroTrackingPos**—Sets the PPX, PPY, PPZ, and PPW system variables based based on the current position.
- 147. **GLType2Begin**—Marks the beginning of a G&L Type 2 command.
- 148. **GLType2CLS**—G&L Type 2 Call subroutine macro. This macro expect at least 1 argument. The first argument must be the subroutine name.
- 149. **GLType2DFS**—G&L Type 2 Define subroutine macro. G&L subroutines can be defined inline, but are not executed until called.
- 150. **GLType2End**—G&L Type 2 End subroutine macro. A number of event can cause the end of a Type 2 command. In order to correctly process a G&L Type 2 file, the Begin/End must be kept in sync.
- 151. **GLType2FXZ**—G&L Type 2 Fixture Offset Zero macro. Expects 1 or 2 arguments. If 1, the specified fixture offset will be reset. If 2, the specified range of fixture offsets will be reset.
- 152. **GLType2FXZArg**—G&L Type 2 FXZ arguments macro. Specifies the arguments for the **GLType2FXZ** macro.
- 153. **GLType2PDO**—G&L Type 2 Probe Offset macro. Arguments are expected in Type I format.
- 154. **GLType2PdId**—G&L Type 2 Probe Offset direction macro.
- 155. **GLType2PGM**—G&L Type 2 Program name macro. This macro requires the program name to be specified as the first argument.
- 156. **GLType2PRB**—G&L Type 2 Probe Datum Offset Activate. This macro requires the probe id to be specified as the first argument.
- 157. **GLType2RPT**—G&L Type 2 Repeat macro. This macro requires the repeat count to be specified as the first argument.
- 158. **GLType2RTA**—G&L Type 2 Return Arguments macro. This causes a return from subroutine and the passing back of specified values to the calling routine. The values that are to be returned are specified as arguments to this macro.
- 159. **GLType2UPD**—G&L Type 2 Update Probe Display macro. This routine is passed a series of arguments that are messages to be display to the console.
- 160. **GotoJump**—Branches to the specified sequence number.
- 161. **GotoJumpForwardBackward**—Similar to **GotoJump** except uses the sign on the value to determine whether it should look forward or backwards from the current position. A negative value will cause it to look backwards. If within a subroutine, the forward and backwards search is limited to the subroutine.
- 162. **GotoLabel**—Branches to the label specified with **LabelName**.
- 163. **Hcode_Tool**—Calls **ToolLengthCompValue**, **ToolCode**, and **ToolChange**. **Note:** This macro is no longer needed and may be deleted in the future.
- 164. **Heid_BlockInit**—Heidenhain specific, block initialization macro.

165. **Heid_CircleCenterX, Heid_CircleCenterY, Heid_CircleCenterZ**—Sets the X,Y,Z coordinates of the center of a circle, respectively. The Circles... modals determine how these values are interpreted.
166. **Heid_EndSub**—Similar to **EndSub**, except a return from the main does not cause an error message.
167. **Heid_MotionLinear**—Similar to **MotionLinear**, except this macro will also reset the circle parameters.
168. **HelicalFullLoops**—Specifies the number of full loops should be processed for a given helical motion. Using this value, the start and end angle, and the helical offset, the pitch is then calculated. The default for this value is 0.0 (generally the pitch if specified, and the number of loops is calculated).
169. **Icode**—Depending on the motion type (circular or non-circular), either sets the X coordinate of a circle, or establishes a 3D offset. The interpretation of the center point is dependent on the Circles control settings.
170. **IfCheck**—Sets an internal variable indicating whether the corresponding “IF” statement was TRUE or FALSE. This variable is reset at the beginning of the next block.
171. **IgnoreMacro**—Ignores the word/value that called it (causes no action). During the creation of a report, the macro will create an unsupported word/value entry.
172. **Ijk2AbcType**—Typically called during the "Start of Processing" event, this macro controls the type of conversion required to interpret IJK rotary motion data for ABC rotations. Use with **[IJK]Vector** macros. Conversion is specific to machine and control, and is specified via **Override Value** as follows:
- 1—Converts IJK rotary data to AB rotations for Forest Line 5 axis mill with A/B rotation and ATEK control. In this case, the macro also rotates the work and shift offsets based on the rotation that is applied to the stock. An assumption is made that the entire amount of the dynamic and work offsets should be rotated by the amount the part is rotated. This logic will not work when the machine is not built about zero, and only part of the work offset is to be rotated.
 - 2—Same as case 1 above, except both A/B rotaries are on the “tool side” of the machine.
- Note:** Rotary control settings should probably be set to "**Absolute**" with the direction set to "**Shortest Distance**".
173. **IncrementalShift**—Incrementally shifts the work coordinate system by the specified XYZ values. Any XYZ value which is not specified is assumed to be zero.
174. **Input**—Renamed to **InputMacro**.
175. **InputMacro**—Stops the processing of the tool path file, and wait for user input.
176. **IVector, JVector, KVector**—Reads the IJK vector values for a rotary motion in IJK format. See also: **Ijk2AbcType**.
177. **Jcode**—Depending on the motion type (circular or non-circular), either sets the Y coordinate of a circle, or establishes a 3D offset. The interpretation of the center point is dependent on the Circles control settings.
178. **Kcode**—Depending on the motion type (circular or non-circular), either sets the Z coordinate of a circle, or establishes a 3D offset. The interpretation of the center point is dependent on the Circles control settings.

- 179.**KtType2CLS**—K&T Type 2 call subroutine macro where the subroutine call record does not have to specify loop count (defaults to 1). The second argument optionally specifies the loop count when it is a single value. Otherwise the second argument does not apply to loop count. The loop count can also be set via variable 0 within the CLS command.
- 180.**Label**—Renamed to **LabelMacro**.
- 181.**LabelMacro**—Identifies the current line with the given Label.
- 182.**LabelName**—Sets the **LabelName**. Used with **GotoLabel**.
- 183.**LoadNextTool**—Loads the next tool listed in the Tool Index table. Tools are retrieved from the table in sequential order. For example, the first tool change loads the first tool in the table for the active subsystem, the second tool change loads the second tool in the table, and so on. This macro is useful to automate tool change activity in VERICUT for manual tool change machines; typically called via the “M 0” group.
- 184.**LoopCount**—Sets the loop count. The loop count is used for the number of times a subroutine is to be executed, and is also used for the number of times the motion in the current block should be repeated. (non-modal)
- 185.**MacroVar**—Fanuc specific macro that establishes machine code file subroutine parameters used with the **CallNCMacro** macro. The mapping of word name to input parameter is: "A" = 1, "B" = 2, ..., "Z" = 26.
- 186.**MahoSequence**—Similar to **Sequence**, except that it also uses the sequence number to recognize the beginning of a new subroutine. Numbers 9000 or above are reserved for program and subroutines. This is MAHO/PHILLIPS specific.
- 187.**MahoVertical2Rest**—Rotates Maho's vertical spindle to the 180 position.
- 188.**MahoVertical2Work**—Rotates Maho's vertical spindle to the 0 position.
- 189.**MaxTableRPM**—Sets the max table RPM's to be the specified value. This is used in turning when calculating the current RPM while in constant surface speed mode.
- 190.**MazakMillingInterpolationOff, MazakMillingInterpolationOn**—Turns off/on a special milling mode for a Mazak controller which causes X values to be multiplied by 2, respectively.
- 191.**MazakPartLoader**—On the first call, it calls **PartTransformation** with a value of 1 and loads the first fixture within VERICUT. On the third call, it calls **PartTransformation** with a value of 2 and loads the second fixture within VERICUT.
- 192.**MazakSpindleRotation**—If in milling mode, the B axis will be rotated to the specified value. In turning mode, the B axis will be rotated to the 0 position.
- 193.**MessageMacro**—Creates a message using the input text string value. The message type is specified by the input numeric value. 1= Error, 2=Warning, 3= Info, 4= Debug, 5= Input, 6=Output, 7= Macro Modal, 10= Error Continuation
- 194.**MirrorA, MirrorB, MirrorC**—Mirrors ABC rotations by negating the ABC values. (Works only for linear tables.)
- 195.**MirrorACancel, MirrorBCancel, MirrorCCancel**—Cancels mirroring by **Mirror[ABC]**.
- 196.**MirrorAValue, MirrorBValue, MirrorCValue**—Same as **Mirror[ABC]**.
- 197.**MirrorCancel**—Cancels all mirroring.

- 198.**MirrorX, MirrorY, MirrorZ**—Mirrors values about the XYZ axes.
- 199.**MirrorXCancel, MirrorYCancel, MirrorZCancel**—Cancels the mirroring about the XYZ axes.
- 200.**MirrorXValue, MirrorYValue, MirrorZValue**—Mirrors values about an axis having the specified XYZ value.
- 201.**ModeAbsolute, ModeIncremental**—Sets the command mode to Absolute or Incremental, respectively.
- 202.**MotionCCW, MotionCW**—Sets the motion type to CCW or CW, respectively.
- 203.**MotionLinear**—Sets the motion type to Linear.
- 204.**MotionNurbs**—Sets the motion type to Nurbs and initialize the nurbs data.
- 205.**MotionOnOff**—Controls when motions and cutting are simulated, depending on values passed to the macro. Calling this macro outputs a “PPRINT/VERICUT-MOTION” command to the Output file.
- 1—Simulates motions (default). This value is automatically set when a reset is done.
- 0—Turns motion simulation and cutting off. Motion commands and positions are processed, but simulation, cutting, collision detection, and axis overtravel detection are turned off during these moves. When motion is turned on, the machine/tool “jumps” to its current position.
- 206.**MotionPlaneXY, MotionPlaneYZ, MotionPlaneZX**—Sets the motion plane to be the XY,YZ,ZX plane, respectively.
- 207.**MotionPoly**—Sets the motion type to Poly.
- 208.**MotionRapid**—Sets the motion type to Rapid.
- 209.**MotionThread**—Sets the motion type to Thread.
- 210.**MotionToolOffsetNeg, MotionToolOffsetPos**—Uses the cutter compensation value and offset the XYZ values in the current block by this amount opposite the direction of motion, or in the direction of motion, respectively.
- 211.**NcodeSeqSub**—Similar to **Sequence**, except that it also marks the sequence number as the beginning of a subroutine during the first pass
- 212.**NULL**—Renamed to **NullMacro**.
- 213.**NullMacro**—Immediately returns-no action is performed. This macro can be used to avoid error messages that are not important for the processing of the program. The macro can also be used in the case of G73X1 (MAHO). In this example, the X1 will cause the mirroring in X to be cancelled (since G73 is also in the block). Since the operation has been executed with the processing of the X value, there is nothing left to be done with the G73 value. G73 can be set to call the “null” macro.
- 214.**NumIfCheck**—Similar to **IfCheck** except if no comparison is present, it will act as an unconditional branch.
- 215.**NumLengthCompX, NumLengthCompY, NumLengthCompZ**—Sets the direction of the tool length compensation.
- 216.**NumPlane**—Defines a motion plane in 3D space. For example: If a circle is drawn in the XY

plane, Z remains constant and only X and Y changes. If a circle is drawn after a 3D plane is established, the circle would be drawn within this 3D plane.

217. **NumPlaneA, NumPlaneB, NumPlaneC, NumPlaneU, NumPlaneV, NumPlaneW**—Defines a 3D plane orientation.

218. **NumSequence**—Depending on the contents of the current block, this macro will execute **Sequence**, **SequenceStartEnd**, or **GotoJump**.

219. **NumSubSequenceLoop**—If “H” is in the block, a call to the corresponding subroutine will be made. Otherwise, the specified sequence of blocks will be executed.

220. **NumSubroutineSequence**—Similar to **SubroutineSequence** except the beginning of a new subroutine will automatically mark the end of the previous subroutine.

221. **NurbsDegree**—Sets the NURBS order to the input value + 1.

222. **NurbsInitialControl**—Adds an initial control point to the current NURBS data.

223. **NurbsKnot**—Initializes the NURBS data (if this defines the beginning of a NURBS statement), and adds a NURBS knot point to the current NURBS data.

224. **NurbsOrder**—Initializes the NURBS data (if this defines the beginning of a NURBS statement), and sets the NURBS order.

225. **NurbsWeight**—Sets the NURBS weight.

226. **Okuma_CallSubName**—Similar to **CallSubName** except the subroutine name begins with the second character.

227. **OkumaCancelShiftRotate**—Cancels both the shift offset and the rotation.

228. **OkumaControllingHead**—Sets the controlling head to be Turret A or Turret B. A value of 110 defines Turret A as the controlling head, and a value of 111 defines Turret B as the controlling head.

229. **OkumaCopyEnd**—Processes the end of the “Copy” command for Okuma. This includes the branching based off of the loop count, and the updating of the shift offset and the rotation value.

230. **OkumaCopyLoopCount**—Sets the Okuma loop count.

231. **OkumaCopyStart**—Processes the start of the “Copy” command for Okuma. This includes the initializing of the branching variables, and the updating of the shift offset and the rotation value.

232. **OkumaShiftRotate**—Sets the Okuma shift offset and rotation values.

233. **Okuma_CallSubName**—Same as **CallSubName**, except the first character of the input text string argument is not included in the name of the subroutine.

234. **OliConvInit**—Special Olivetti conversion initialization routine. This routine is to be called along with the standard **ConversionInit** routine during initialization.

235. **OliLabel**—Olivetti specific routine for handling labels.

236. **OliType2EPP**—Olivetti specific routine for handling the TYPE II EPP command. This command takes 2 label arguments, where to jump to, and when to jump back.

237. **Optimizable**—Increments a counter each time it is called on a block. The counter is used by OptiPath to determine if a block is optimizable, even if G-codes exist in the block that are not

typically optimizable. The block is optimizable when the number of objects in a block equals the optimizable count. The optimizable count is re-initialized with a new block.

238. **OptipathExpandSubs**—When called with an **Override Value** of "1", then MCD subroutines will be optimized and expanded. "0" does not optimize or expand MCD subroutines. To optimize the main program as well as expand and optimize subs, call this macro via the "Start of Processing" event with an **Override Value** of "1".
239. **PartNormalX, PartNormalY, PartNormalZ**—Used with **Tool3dOffset** to define the normal to the part surface.
240. **PartTransformation**—Uses the Part Transformation table to send a MATRIX/REVERSE command to vericut. The values in the table are: rotate x, rotate y, rotate z, translate x, translate y, translate z, reverse flag.
241. **PolarInterpolation**—Turns on and off polar interpolation. 1 for on, 0 for off.
242. **PolarLinearAxis**—Sets the linear axis that will driven based on the cartesian coordinates. The data is passed as either a 1, 2, or 3 (X, Y, Z). The default is 1.
243. **PolarRotationalAxis**—Sets the rotational axis that will driven based on the cartesian coordinates. The data is passed as either a 4, 5, or 6 (A, B, C). The default is 6.
244. **PolarXValue, PolarYValue, PolarZValue**—Sets the XYZ cartesian coordinate values for Polar Interpolation.
245. **ProcessCompNameValue**—Sets the position of the component whose name is specified in the input text string to the specified numeric value.
246. **ProcessCompRegValue**—Sets the poistion of the component whose register value is specified in the input text string to the specified numeric value.
247. **ProcessMotion**—Processes the motion associated with the previous set of commands. This command is automatically called at the end of the block. This command can also be called as a macro to create additional motion.
248. **ProgramStartNextLine**—Marks the following block as the first block of the program. This is used when the beginning of a file contains subroutines followed by a specific code which marks the beginning of the program.
249. **Rcode**—Depending on the motion type (circular or non-circular), either sets the radius of a circle, or sets the R point level of a cycle. For a circle the interpretation of the center point is dependent on the Circles control settings.
250. **ReferencePoint**—Causes a return to the machine reference point for the axes specified on the block. This action occurs after block motion has been processed.
251. **ReferencePointCancelOffsets**—Causes a return to the machine reference point for the XYZ axes specified on the block. This action occurs after block motion has been processed. The corresponding work and shift offsets for the specified axis are also reset to 0.
252. **ResetWorkCoordParms**—Resets the work coordinate parameters.
253. **RestoreMotionType**—Restores the motion type to the last non-rapid motion type
254. **RestoreUnits**—Restores the units to the previous saved value. See **SaveUnits**.
255. **RetractLevelInitial**—Sets the retract level to be the initial level at the point at which the cycle

mode was turned on.

- 256.**RetractLevelRpoint**—Sets the retract level to the R point level.
- 257.**RetractLevelSpecifiedpoint**—Sets the retract level to a specified point. This point is specified with either **[XYZ]RetractSpecifiedpoint** macros.
- 258.**RotaryDirCCW, RotaryDirCW**—Sets the direction for a **EIA (360 Absolute)** rotary table to be always Counter Clockwise or Clockwise, respectively. By default, the affect applies to all rotary axes. Individual rotary axes can be controlled via entering the axis name in the **Override Text** field, e.g. “AAXIS”, “BAXIS”, etc.
- 259.**RotaryDirLinear**—Sets the direction for a **EIA (360 Absolute)** rotary table to be linear. The value will be first converted into a number between 0 and 360, and then the rotation will act as if it is a linear table. By default, the affect applies to all rotary axes. Individual rotary axes can be controlled via entering the axis name in the **Override Text** field, e.g. “AAXIS”, “BAXIS”, etc.
- 260.**RotaryDirPosCCW**—Sets the direction for a **EIA (360 Absolute)** rotary table to be Counter Clockwise when the angle is positive and Clockwise when the angle is negative. By default, the affect applies to all rotary axes. Individual rotary axes can be controlled via entering the axis name in the **Override Text** field, e.g. “AAXIS”, “BAXIS”, etc.
- 261.**RotaryDirPosCW**—Similar to **RotaryDirPosCCW**, but rotary directions are reversed.
- 262.**RotaryDirShortestDist**—Sets the direction for a **EIA (360 Absolute)** rotary table to be the direction which is the shortest distance from the current position. By default, the affect applies to all rotary axes. Individual rotary axes can be controlled via entering the axis name in the **Override Text** field, e.g. “AAXIS”, “BAXIS”, etc.
- 263.**RotaryLinearRewind**—Converts the current positions of the ABC Rotary Linear tables to an absolute value between 0-360. No motion takes place.
- 264.**Rotate**—Renamed to **RotateMacro**.
- 265.**RotateCancel**—Cancels the current rotation.
- 266.**RotateMacro**—Fanuc specific rotation that causes the current coordinate system to be rotated by a specified angle. The angle is specified with **RotationAngle** . The actual point of rotation will be dependent on whether the motion which follows is in incremental or absolute mode. Rotate will apply the total rotation to the current working coordinate system (which might already be rotated). The total rotation value is therefore actually an incremental rotation value.
- 267.**RotateOrigin**—Rotates the working coordinate system by the total rotation angle. The rotation is always done about origin within the working coordinate system. This is not an incremental rotation.
- 268.**RotateXValue, RotateYValue, RotateZValue**—Sets the XYZ coordinate values of the point of rotation.
- 269.**RotateXYZ**—Rotates the working coordinate system about the specified position. Then angle can be specified with either **RotationAbsAngle** or **RotationIncAngle**.
- 270.**RotationAbsAngle**—Sets the total rotation angle to be the specified value.
- 271.**RotationAngle**—Similar to **RotationAbsAngle**, but is typically used with Fanuc and with **RotateMacro**.
- 272.**RotationIncAngle**—Increments the total rotation angle by the specified value.

273. **RPM**—Renamed to **RPMSpeed**.
274. **RPMMode**—Sets the current mode to constant RPM's.
275. **RPMSpeed**—Sets the spindle speed in RPM's.
276. **RtcpOff**, **RtcpOn**—Turns RTCP processing off or on, respectively. **Note:** After turning RTCP on, the **UpdateRotaryOffsets** macro should probably be called to make sure that all internal offsets are updated accordingly. This macro should be called after RTCP has already been turned on. **UpdateRotaryOffsets** should probably be executed after motion or a block by itself.
277. **SaveUnits**—Saves the current Units settings. See also **RestoreUnits**.
278. **Sequence**—During pass 1, updates the sequence table and set the current sequence number.
279. **SequenceEnd**—Specifies the ending sequence number within a sequence loop.
280. **SequenceLoop**—Executes the sequence loop based on previously set parameters. If the loop count was not specified, 1 is assumed. If the ending sequence number was not specified, the starting sequence number is assumed.
281. **SequenceLoopCount**—Specifies the number of times to execute the sequence loop.
282. **SequenceStart**—Specifies the starting sequence number within a sequence loop.
283. **SequenceStartEnd**—Defines the start and ending sequence number within a sequence loop. This macro is called twice within a block. On each call, either the start or ending sequence number is passed. The starting sequence number is defined to be the lower of the two numbers.
284. **SetAbsWorkCoord**—Similar to **SetAdditionalWorkCoord**, except the offsets are always interpreted as absolute values (not incremental).
285. **SetActiveSpindleName**—Specifies the Active Spindle Name for a subsystem. This macro is typically used to change where material removal occurs when simulating pick-off spindle operations.
286. **SetAdditionalWorkCoord**—Sets entries 0-59 in the Work Offsets table. The work coordinate index must be between 0-59. The corresponding entry in the table are then updated based on the specified XYZ values. If an XYZ value is not specified, the corresponding entry is not updated.
287. **SetBaseWorkOffset**—Sets a base offset that is added to any shift or work coordinate offsets. The base offset is retrieved from the Base Work Offset table.
288. **SetIncWorkCoord**—Similar to **SetAdditionalWorkCoord**, except the offsets are always interpreted as incremental values (not absolute).
289. **SetPivotOffset**—Sets the pivot offset for a tool component connected to a rotary component. This macro should be passed the index into the RTCP Pivot Offset table. If the entry in the table does not exist, the system automatically calculates the Z offset value using the distance from the Tool component to the pivot point of the rotary axis to which the Tool component is connected (X and Y offsets set to 0). If the tool axis and the rotary axis are collinear, then the RTCP Pivot Offset will be set the to difference between the Tool and the Rotary component origins.
- Note:** If a spindle changer is being used for a job in Tool Tip mode and using RTCP, then the RTCP Pivot Offset table needs to be set whenever a new spindle is attached.
290. **SetSubsystemID**—Sets the current subsystem to to the input text string value.

291. **SetTableCoord**—The text string contains the table name to be set. This macro sets the entry (specified by **TableIndex**) in this table to the values specified with the **WorkCoord[ABCXYZUVW]Value** macros. Nine values will be stored in the table. Any value not set in the block will assume to be zero.
292. **SetTableValues**—The text string contains the table name to be set. This macro sets the entry (specified by **TableIndex** in this table) to the values specified in **TableValues**. The order of the values passed to **TableValues** will be the order stored in the table. The number of values stored will be equal to the number of times **TableValues** was called on this block.
293. **SetToolNumVar**—When passed an override text value of the variable to be set, this macro will set this variable to be the current tool number. Typically, this would be called immediately after **ToolChange**.
294. **SetWorkCoord**—Sets entries 53-59 in the Work Offsets table. The work coordinate index must be between 0-6. The corresponding entry in the table are then updated based on the specified XYZ values. If an XYZ value is not specified, the corresponding entry is not updated.
295. **Siemens3CmdBranch**—The value passed is the sequence number to branch to. If an additional argument is passed, then it is added to the absolute value of the sequence number. This routine then calls the **GotoJumpForwardBackward** macro. If more then 2 arguments were passed, then an error message will be printed.
296. **Siemens3CmdIfEqual**—The value passed is the sequence number to branch to. If argument 1 is equal to argument 2, then **GotoJumpForwardBackward** is called to branch to the specified sequence number. If an additional argument is passed, then it is added to the absolute value of the sequence number. If less then 2 arguments, or more then 3 arguments were passed, then an error message will be printed.
297. **Siemens3CmdIfGreater**—The value passed is the sequence number to branch to. If argument 1 is greater then argument 2, then **GotoJumpForwardBackward** is called to branch to the specified sequence number. If an additional argument is passed, then it is added to the absolute value of the sequence number. If less then 2 arguments, or more then 3 arguments were passed, then an error message will be printed.
298. **Siemens3CmdIfGreaterEqual**—The value passed is the sequence number to branch to. If argument 1 is greter then or equal to argument 2, then **GotoJumpForwardBackward** is called to branch to the specified sequence number. If an additional argument is passed, then it is added to the absolute value of the sequence number. If less then 2 arguments, or more then 3 arguments were passed, then an error message will be printed.
299. **Siemens3LoadAddress**—Sets the contents of the address parameter specified by **Siemens3SetAddressParm** to the specified value. If a variable argument is specified, it will override the input value. A value of 1 corresponds to the X Axis, a value of 2 corresponds to the Z axis.
300. **Siemens3PreProcessor**—Controls when a Siemens 3T PreProcessor translates blocks of Siemens 3T type data into a more standard form. This preprocessor is specific to the Siemens 3T syntax, and should not be usedwith the Siemens 880 or the Siemens 7M controllers. The preprocessor will be activated after reading a block from the MCD file, and before substitutions.

This macro should be passed a value of 0 or 1. A value of 0 turns off the Siemens 3 preprocessor, a value of 1 turns on the Siemens 3 preprocessor. If used, this macro should be called during the NEW "Reset" event.

The following conversions are made by the preprocessor:

R20 200 R30 300 => R20=200 R30=300

R30 R20 => R30=R30+R20

R30-R20 => R30=R30-R20

R30/R20 => R30=R30/R20

R30.R20 => R30=R30*R20

R305 => R30 =5

R30.5 => R30 =.5

Z1.6 R07 => Z1.6+R07

R30 0 R20 => R30=R20

R10 R20.R30 => R10=R10+R20 R20=R20*R30

X1.5 R10 R20 => X1.5+R10 R10=R10+R20

X R10 R20 => X R10 R10=R10+R20

Note: OptiPath is not supported with the Sinumerik 3T control.

301. **Siemens3SetAddressParm**—Sets the address parameter that will be accessed (90-93). This is necessary since @90 command might be followed by 2 values (@90 0 R10). This allows for @90 to be defined as a word. This word can then have a value and an optional variable argument.
302. **Siemens3TAddressAxis**—Calls the corresponding "AxisMotion" macro with the specified value. If a variable argument is specified, it will override the input value. The axis to move is specified by contents of the specified address parameter. The address parameter is set using the **Siemens3SetAddressParm** macro.
303. **SiemensCmdAbs**—Take the absolute value of the second argument and stores the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
304. **SiemensCmdAngle**—Takes the arc tangent of y/x (second/third arguments), and stores the result in degrees in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
305. **SiemensCmdASin**—Takes the arc sine of the second argument, and stores the result in degrees in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
306. **SiemensCmdATan**—If 2 arguments were passed, takes the atan of the second argument, converts the results to degrees, and stores the result in the first argument. If only one argument was passed, takes the atan of the first argument, converts the results to degrees, and stores the result in the first argument. If no arguments or more than 2 arguments were passed, issues an error message.
307. **SiemensCmdBranch**—Branches to the location specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
308. **SiemensCmdCase**—Processes a Siemens "Case" statement. The first argument is the value being tested. It is followed by a series of value pairs (value, branch location). If the values match, a branch is made to the specified location. Arguments are set using **ValueArgument** and

VariableArgument.

309. **SiemensCmdCos**—Takes the cosine of the second argument (in degrees), and store the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
310. **SiemensCmdDec**—Decrements the value of the specified variable. Arguments are set using **VariableArgument**.
311. **SiemensCmdHypot**—Takes the square root of the sum of the squares of the second and third arguments, and store the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
312. **SiemensCmdIfEqual, SiemensCmdIfGreater, SiemensCmdIfGreaterEqual, SiemensCmdIfLess, SiemensCmdIfLessEqual, SiemensCmdIfNotEqual**—These macros process Siemens conditional logic. They are in the form: IF A <conditional> B THEN continue ELSE jump to C. Arguments are set using **ValueArgument** and **VariableArgument**.
313. **SiemensCmdInc**—Increments the value of the specified variable. Arguments are set using **VariableArgument**.
314. **SiemensCmdIntegral**—Truncates the decimal contents of the specified variable. Arguments are set using **VariableArgument**.
315. **SiemensCmdSin**—Takes the sine of the second argument (in degrees) and stores the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
316. **SiemensCmdSquareRoot**—Takes the square root of the second argument and stores the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
317. **SiemensCmdTan**—Takes the tangent of the second argument (in degrees) and stores the result in the variable specified by the first argument. Arguments are set using **ValueArgument** and **VariableArgument**.
318. **SiemensCornerValue**—Sinumerik 3T version of the **CornerMode/CornerValue** macros. A negative value sets the Mode to CHAMFER, and a positive value set the mode to RADIUS. The value also defines the **CornerValue** as described for "**CornerValue**".
319. **SiemensNurbsKnot**—Adds a Siemens specific incremental knot value.
320. **SiemensNurbsProcess**—Siemens specific. Checks to see if the NURBS statement has been completed, and if so, processes the NURBS statement..
321. **SiemensPlaneSelection**—Resets the plane selection argument count to zero.
322. **SiemensPlaneSelectionArg**—Processes the arguments for the Siemens G16 command. The first 2 parameters set the motion plane. The third parameter specifies the axis in which the tool length is to be applied, and the fourth parameter specifies whether the additional offset is to be applied to either the tool length (tool length comp) or the cutter comp.
- This functionality is provided with the exception that the tool length (gage offset) is always applied in the direction of the tool at the time the offset is applied (typically tool change), and there are still unanswered questions about how the fourth argument is applied.
323. **SiemensPolyT**—Set the POLY T value.

324. **SiemensPolyx**—Sets the X axis, ax, and the bx values based on the Siemens specific input text string argument.
325. **SiemensPolyy**—Sets the Y axis, ay, and the by values based on the Siemens specific input text string argument.
326. **SiemensPolyz**—Sets the Z axis, az, and the bz values based on the Siemens specific input text string argument.
327. **SiemensShiftAngleA, SiemensShiftAngleB**—Specifies the absolute rotation angle. The sum of Angle A and Angle B is the total rotation angle that will be applied.
328. **SiemensShiftOffsetA, SiemensShiftOffsetB**—Similar to **AbsoluteShift**, except two different shift offsets are supported. The total shift offset is the sum of A and B.
329. **SiemensSubroutineCallDefine**—Processes the Sinumerik L word as described in the Siemens Controller Functions document. If the last 2 digits are zero, the L word is being used to define a subroutine. If it is non-zero, the last 2 digits represents the number of times the specified subroutine should be called. The remaining portion of the value represents the subroutine name. Also, the **Sequence** macro is automatically called to update the sequence table and set the current sequence number.
330. **SpindleChanger**—Unloads the current spindle (if one exists), and loads the spindle specified with the **SpindleValue** macro. This macro is a MasterCenter specific macro.
331. **SpindleMotionCCW, SpindleMotionCW**—If the spindle speed was set, outputs the APT "SPINDL/%n,CCLW" or "SPINDL/%n,CLW" statement, respectively. Otherwise, the APT "SPINDL/ON" statement is output.
332. **SpindleMotionOff**—Outputs the APT "SPINDL/OFF" statement.
333. **SpindleSpeed**—Sets the spindle speed to the input value.
334. **SpindleSpeedMax**—Sets the maximum spindle speed to the input value.
335. **SpindleSpeedMin**—Sets the minimum spindle speed to the input value.
336. **SpindleValue**—Sets the spindle value which is to be loaded.
337. **StopOptional**—Outputs the APT "OPSTOP" statement.
338. **StopProgram**—Outputs the APT "STOP" statement.
339. **SubroutineName**—Sets the subroutine name that is then used by the **CallNCMacro** or **CallSub** macros.
340. **SubroutineSequence**—Similar to **Sequence**, if the job is defined as a single pass job. On multi-pass jobs, the current location within the file is marked during the first pass as the beginning of a subroutine.
341. **SubStartingSeq**—Sets the starting sequence number for a call into a subroutine.
342. **SubsystemActivateTool**—Activates the tool associated with the specified subsystem. The subsystem is specified with the input text string argument.
343. **SurfaceSpeed**—Specifies that the part spindle RPM's will be set such that the surface speed of the tool will remain constant at the given rate. The rate is specified as Feet Per Minute or Meters Per Minute.

344. **Sync**—Renamed to **SyncMacro**.
345. **SyncMacro**—Causes the active head to wait at the current location until the other head reaches the corresponding sync location.
346. **SyncTransfer**—Causes the controlling/slave heads to swap
347. **SyncValue**—Sets the current Sync value. This will cause the current head to wait until the other processes a Sync value that is greater then or equal to the specified value.
348. **TableIndex**—Sets the index/register number of the table entry to be modified. Used with **SetTableValues** and **SetTableCoord**. Default is 1.
349. **TableValues**—Can be called up to nine times to store nine values. Used with **SetTableValues**.
350. **ThreadWire**—Thread the wire EDM wire. This will cause the Wire to be displayed within the simulator, and a “PPRINT/VERICUT-TC” statement to be printed. The height for the cutter will be set to the maximum length of the wire based off of the vertical distance between the guides, and the maximum angle.
351. **Tool3dOffset**—Turns on 3D Tool Correction to keep the tool tangent with the cutting surface. The actual tool offset will vary based on the vector which is normal to the part surface at each given point. This type of tool offset can only be used with standard bull nose and ball nose cutters.
352. **Tool3dXOffset, Tool3dYOffset, Tool3dZOffset**—Sets the 3D offset in the XYZ directions, respectively. This is a form of cutter compensation where the offset can be in 3D, and is specified directly in the MCD file. **Note:** These macros are not related to **Tool3dOffset**.
353. **ToolChange**—When the modal **Tool Number Method=Select Only**, unloads the current tool (if one exists) and load the new tool.
354. **ToolChangeIfDifferent**—Calls the **ToolChange** macro only when the new tool number is different from the current tool.
355. **ToolChangeNoRetract**—Similar to **ToolChange**, except that the machine components will not retract to their tool change locations.
356. **ToolChangeSave**—Causes the previously saved tool number to be loaded. See **ToolCodeSave**.
357. **ToolCode**—When the modal **Tool Number Method=Select Only**, set the new tool number to the input value. But, when **Select & Change** is the active choice, this macro set the new tool number to the input value and executes the corresponding tool change.
358. **ToolCodeSave**—Saves the current tool number. See **ToolChangeSave**.
359. **ToolLengthCompNeg**—Sets the tool length offset mode to Negative.
360. **ToolLengthCompOff**—Turns off the tool length offset.
361. **ToolLengthCompPos**—Sets the tool length offset mode to Positive.
362. **ToolLengthCompToolNum**—Uses the current tool number to index into the Tool Length Compensation table. The corresponding value is used as the compensation amount when Tool Length Compensation is active.
363. **ToolLengthCompUpdate**—Updates the Tool Length Compensation amount. It assumes that Tool Length Comp (pos/neg) has previously been turned on.

364. **ToolLengthCompValue**—Uses the input value to index into the Tool Length Compensation table. The corresponding value is used as the compensation amount when Tool Length Compensation is turned on.
365. **ToolLengthOffsetValue**—Sets the tool offset value based off of the corresponding entry in the Tool Length Compensation table. This value is modal.
366. **ToolNoseCompApply**—Causes tool nose compensation to be applied immediately.
367. **ToolNoseCompCancel**—Turns off tool nose compensation.
368. **ToolNoseCompValue**—Specifies the index into the Tool Nose Compensation table, which then specifies the xyz tool nose compensation that is to be applied. These values are typically only active while in cutter compensation mode.
369. **ToolOffsetAptAdj**—VERICUT workpiece view does not support the concept of multiple driven points for a given tool. However, machine views do. This macro allows Gage Offset table to drive a secondary offset on the tool, while the APT is still based on the driven point defined by the Tool Library file. While generating the APT, the Gage Offset is calculated by: GAGEPOINT_OFFSET - DRIVENPOINT_OFFSET.
370. **ToolOffsetAptAdj2**—Similar to **ToolOffsetAptAdj**, except is used with VERICUT to adjust for the cutter origin, which is not currently used by VERICUT.
371. **ToolOffsetIndex**—Specifies the entry in the Gage Offset table that is to be used to specify the distance from the driven point of the tool to the spindle face. **ToolOffsetUpdate** will need to be called to actually update the offset.
372. **ToolOffsetUpdate**—Updates the gage offset based off of the value specified in the **ToolOffsetIndex** macro.
373. **ToolOffsetUpdate2**—Similar to **ToolOffsetUpdate**, except is used with VERICUT to adjust for the cutter origin, which is not currently used by VERICUT.
374. **ToolVectorX, ToolVectorY, ToolVectorZ**—Sets the tool vector. Is specified with **ToolChange**.
Note: These macros are redundant and may be deleted in the future.
375. **TurretActivateTool**—Marks the tool associated with the specified tool index as the active tool.
Note: APT output is generated from the active tool.
376. **TurretIndex**—Rotates the turret so that the specified tool index number is in cutting position. This macro will then search for a tool with the specified index number (index number only applies to tools that are connected to a turret). If an entry exists for the specified tool index in the Turret Rotations table, the corresponding value will be used as the absolute angle of this tool. If an entry does not exist, the angle will be calculated based off of the XAXIS/ZAXIS matrix for the associated tool.
- The register name of the corresponding turret will be used to rotate the turret into position. The Rotary control settings determine the direction (CW,CCW) of motion. Typically this should be set to **Shortest Distance**. While the turret is being rotated, no APT will be generated. An APT GOTO will be generated at the final position.
377. **TurretLoadTool**—For a turret, a tool is not actually loaded. This macro is used for turret machines to reset the cycles (modal dependent), reset the motion type (modal dependent), and issue a LOADTL statement.
378. **TurretRetract**—Causes the Turret to retract to the tool_change retract position. All corresponding tool change modals/table values are used to determine if the retract should actually

occur, which components should retract, and the position of the retract.

379. **Type2BEQLabel**—This is a Type 2 macro which expects three arguments. If the first 2 arguments are equal the processing will jump to the label specified by the third argument.
380. **Type2CLS**—A generic Type II subroutine call with no arguments.
381. **Type2DFS**—A generic Type 2 subroutine definition with the ability to jump over the subroutine if it is being defined and not called. This macro should be passed one argument which is the subroutine name. Additional arguments are ignored.
382. **Type2Goto**—This is a Type 2 macro which expects one argument. This argument can either be a label or a sequence number (a label will take precedence). This macro executes a branch to the specified location.
383. **Type2If**—This is a Type 2 macro which expects one argument. If the argument equates to TRUE, the corresponding statement will be executed. This routine might be G&L specific.
384. **Type2Label**—This is a Type 2 macro which expects one argument. The argument is the label name that is to be applied to the current block.
385. **Type2OpMsg**—This is a Type 2 macro which expects one argument. The argument is the operator message that is to be printed to the console.
386. **Type2OpMsgAllArgs**—This is a Type 2 macro which can except any number of arguments. The text string associated with each argument will be concatenated together and will be printed as an INFO message.
387. **Type2OpMsgStop**—This is a Type 2 macro which has one optional argument. The argument is the operator message that is to be printed to the console. This macro will also cause the processing of the MCD file to stop.
388. **UAxisMachineMotion**—Sets the U value used to process the machine's U component. This macro is similar to **UAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
389. **UAxisMotion**—Sets the U value used to process the machine's U component.
390. **Unitize3DVector**—Sets a flag to determine whether the vector associated with 3D tool offsets should be unitized or not. 0= No, 1= YES.
391. **UnitsInch, UnitsMetric**—Sets the units mode to Inch or Metric, respectively.
392. **UnloadTool**—Unloads the current tool.
393. **UnloadToolNoRetract**—Similar to **UnloadTool**, except that the machine components will not retract to their tool change locations.
394. **Unsupported**—Prints a warning message that the specified word/value pair is unsupported.
395. **UpdateRotaryOffsets**—Updates rotary pivot offset values that may have changed due to turning RTCP on/off.
396. **UpdateIPZ**—This macro is passed an index into the Input Program Zero table via the **Override Value** field. Use this macro to specifically set the internal values for the Input Program Zero offset; a capability that is mainly useful when working with a “spindle changer” type of machine and running in Tool Tip mode. **Note:** this does not change any of the table values. The new value for IPZ can be seen using Modals/Debug/Debug Offsets.

Note: This macro should probably be executed after motion or on a block by itself. Also, since the internal value for Input Program Zero is being modified, this method will probably not work when running in Tool Length Comp mode.

397. **uvw_axis**—Renamed to **UvwAxis**.
398. **UvwAxis**—Cutter Compensation can only be applied along the x,y, and z axis. In the case of a dual head system where xyzuvw all exists, cutter compensation can only be applied along the xyz or uvw axis at a given time. This macro activates the uvw axis and deactivates the xyz axis.
399. **ValueArgument**—Specifies a value that is to be added to the list of value arguments.
400. **VariableArgument**—Specifies a variable that is to be added to the list of variable arguments. The contents of the variable is also added to the list of value arguments.
401. **VAxisMachineMotion**—Sets the V value used to process the machine's V component. This macro is similar to **VAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
402. **VAxisMotion**—Sets the V value used to process the machine's V component.
403. **VC_LoadFix**—Outputs a “DISPLAY_FIXTURES” PPRINT.
404. **VC_ModeMilling, VC_ModeTurning**—Outputs a special APT PPRINT statement which tells Vericut to switch to Milling or Turning mode, respectively.
405. **VC_UnloadFix**—Outputs a “REMOVE_FIXTURES” PPRINT.
406. **VoltageOff, VoltageOn**—Controls when a SPINDLE/OFF or SPINDLE/ON statement is output during wire EDM machining, respectively.
407. **WAxisMachineMotion**—Sets the W value used to process the machine's W component. This macro is similar to **WAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
408. **WAxisMotion**—Sets the W value used to process the machine's W component.
409. **WAxisZeroTracking**—Sets the W value used to process the machine's W component on a Zero-Tracking machine.
410. **WhileLoop**—Sets the while condition to the input value.
411. **WorkCoord**—Updates the work coordinate system based off of the values in the Work Offsets table. The input value is used as an index into this table. Valid values are 0-59.
412. **WorkCoordAValue, WorkCoordBValue, WorkCoordCValue**—Sets the ABC values that are to be used with various Work Coordinate System Offset macros, respectively.
413. **WorkCoordIndex**—Set the index into the Work Coordinate Offset values. Valid values are from 0 to 59.
414. **WorkCoordUValue, WorkCoordVValue, WorkCoordWValue, WorkCoordXValue, WorkCoordYValue, WorkCoordZValue**—Sets the UVWXYZ values used with various Work Coordinate System Offset macros, respectively.
415. **WPrimaryTrackingAxis**—Marks the W as the primary axis and the Z as the secondary axis on a zero tracking machine.

416. **XAxisIncrMotion**—Sets the X value used to process the machine's X component. This macro is similar to **XAxisMotion** except it ignores the Absolute/Incremental mode setting, and increment by the value specified.
417. **XAxisMachineMotion**—Sets the X value used to process the machine's X component. This macro is similar to **XAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
418. **XAxisMotion**—Sets the X value used to process the machine's X component.
419. **XLeadRate**—Sets the distance to move along the X axes per revolution while threading. If 2 or more axes are in motion, the lead rate of the axis which is moving the greatest distance will be used.
420. **XRetractSpecifiedpoint**—Sets the retract level X position.
421. **xyz_axis**—Renamed to **XyzAxis**.
422. **XyzAxis**—Cutter Compensation can only be applied along the x,y,and z axis. In the case of a dual head system where xyzuvw all exists, cutter compensation can only be applied along the xyz or uvw axis at a given time. This macro activates the xyz axis and deactivates the uvw axis.
423. **YAxisIncrMotion**—Set the Y value used to process the machine's Y component. This macro is similar to **YAxisMotion** except it ignores the Absolute/Incremental mode setting, and increment by the value specified.
424. **YAxisMachineMotion**—Sets the Y value used to process the machine's Y component. This macro is similar to **YAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
425. **YAxisMotion**—Sets the Y value used to process the machine's Y component.
426. **YLeadRate**—Sets the distance to move along the Y axes per revolution while threading. If 2 or more axes are in motion, the lead rate of the axis which is moving the greatest distance will be used.
427. **YRetractSpecifiedpoint**—Sets the retract level Y position.
428. **ZAxisIncrMotion**—Sets the Z value used to process the machine's Z component. Similar to **ZAxisMotion**, except it ignores the Absolute/Incremental mode setting, and increment by the value specified.
429. **ZAxisMachineMotion**—Sets the Z value used to process the machine's Z component. Similar to **ZAxisMotion**, except it ignores all transformations and is used to move a component to an absolute position within the machine coordinate system.
430. **ZAxisMotion**—Sets the Z value used to process the machine's Z component.
431. **ZAxisMotionNoCycle**—Same as **ZAxisMotion**, except the value is never interpreted as a cycle depth value.
432. **ZAxisZeroTracking**—Sets the Zvalue used to process the machine's W component on a Zero-Tracking machine.
433. **ZLeadRate**—Sets the distance to move along the Z axes per revolution while threading. If 2 or more axes are in motion, the lead rate of the axis which is moving the greatest distance will be used.

434. **ZPrimaryTrackingAxis**—Marks the Z as the primary axis and the W as the secondary axis on a zero tracking machine.
435. **ZRetractSpecifiedpoint**—Set the retract level Z position.
436. **ZToolOffsetCompName**—Specifies the name of the component to which the “Z tool offset” is applied.
437. **ZTranslateStockComp**—Specifies the name of a Stock component in which the turning "PROFILE" primitive is to be translated. This macro is typically used when simulating pick-off spindle operations.
438. **ZTranslateValue**—Specifies the "Z" amount that a turning "PROFILE" primitive is to be translated. This macro is typically used when simulating pick-off spindle operations.

Functions - listed alphabetically

The functions which can be referenced via **Setup menu > Control > Word Format** are listed below in alphabetical order. The following function types are supported:

Mathematical functions—typically passed a single numerical value, and returns a numerical value

Conditional functions—passed the text string of the current block, up to and including the current word, and returns a word type specifying how the current word is to be interpreted (typically identified by the characters “Cond” in the function name)

1. **abs(value)**—Returns the absolute value of the specified value.
2. **acos_d(value)**—Returns the arc cosine 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 0.0 to 180.0.
3. **acos_r(value)**—Returns the arc cosine of the specified value. The specified value must be in the range of -1.0 to 1.0. The return value is in radians, and is in the range of 0 to pi.
4. **asin_d(value)**—Returns the arc sine 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 -90.0 to 90.0.
5. **asin_r(value)**—Returns the arc sine of the specified value. The specified value must be in the range of -1.0 to 1.0. The return value is in radians, and is in the range of -pi/2 to pi/2.
6. **atan_d(value)**—Returns the arc tangent 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 -90.0 to 90.0.
7. **atan_r(value)**—Returns the arc tangent of the specified value. The specified value must be in the range of -1.0 to 1.0. The return value is in radians, and is in the range of -pi/2 to pi/2.
8. **atan2_d(yvalue, xvalue)**—Returns the arc tangent of yvalue/xvalue. The return value is in degrees, and is in the range of -180.0 to 180.0. If yvalue and xvalue are both zero, the return value will be zero.
9. **atan2_r(yvalue, xvalue)**—Returns the arc tangent of yvalue/xvalue. The return value is in radians, and is in the range of -pi to pi. If yvalue and xvalue are both zero, the return value will be zero.
10. **AtanCondDivide(block_str)**—If the ‘/’ symbol is part of a ATAN argument and its is not imbedded within brackets ([]), then this function returns the type SEPARATOR. Otherwise this function returns the type DIVIDE.
11. **ceil(value)**—Returns the least integral value greater than or equal to the specified value.
12. **cosh_d(angle)**—Returns the hyperbolic cosine of the given angle specified in degrees.
13. **cosh_r(angle)**—Returns the hyperbolic cosine of the given angle specified in radians.
14. **cos_d(angle)**—Returns the cosine of the given angle specified in degrees.
15. **cos_r(angle)**—Returns the cosine of the given angle specified in radians.

16. **exp(power)**—Returns the natural logarithm ‘e’ raised to the specified power.
17. **FadalCondAsteriskWord(block_str)**—If the Asterisk Word comes after a “#” character, then this function returns the type MULTIPLY. Otherwise this function returns the type COMMENT.
18. **FadalCondEqualWord(block_str)**—If the Equal Word comes after the string “IF”, then this function returns the type EQ. Otherwise, this function returns the type ASSIGNMENT.
19. **FadalCondLParenWord(block_str)**—If the Left Parenthesis Word comes after a “#” character, then this function returns the type LPAREN. Otherwise this function returns the type COMMENT.
20. **FadalCondNWord(block_str)**—If the N Word comes after a “#” character, then this function returns the type IGNORE. Otherwise this function returns the type COMMAND.
21. **FadalCondRParenWord(block_str)**—If the Right Parenthesis Word comes after a “#” character, then this function returns the type RPAREN. Otherwise this function returns the type END_COMMENT.
22. **FadalCondRWord(block_str)**—If the R Word comes after a “G” character, then this function returns the type COMMAND. Otherwise this function returns the type VARIABLE_TAG.
23. **floor(value)**—Returns the greatest integral value less than or equal to the specified value.
24. **get_current_N_seqno()**—Returns value of the current N word. If the N word was not specified on the current line, the value of the last specified N word is returned.
25. **GLCondPWord(block_str)**—If the P word is part of a PDO command, in CYCLE mode, or if the next non-space character on the block is a “=”, then this function returns the type COMMAND. Otherwise this function returns the type VARIABLE_TAG.
26. **ln(value)**—Returns the natural logarithm of the specified value. The specified value must be positive otherwise zero is returned.
27. **IntTrunc(value)**—Returns the integer portion of the specified value.
28. **KtCondLeftParen()**—This function supports the “(“ word when it is used as mathematical precedence and as Begin Type 2 data.
29. **KtCondRightParen()**—This function supports the “)” word when it is used as mathematical precedence and as End Type 2 data.
30. **log(value)**—Returns the base 10 logarithm of the specified value. The specified value must be positive otherwise zero is returned.
31. **NumAtan(value)**—Similar to atan2_d(yvalue, xvalue), except the return value is in thousandths of a degree.
32. **NumEFunc(value)**—If the variable is being assigned to a COMMAND word, a “NUM” specific factor is applied to the value of the variable..
33. **NumInput()**—Causes the program to prompt the user for input. If the first character of input is a-z or A-Z, a value of 1-26 is returned. If the user enters a numeric value, the value entered will be returned.
34. **NumCondEqualWord(block_str)**—If the Equal Word comes after the string “G79” or the string “IF”, then this function returns the type EQ. Otherwise, this function returns the type ASSIGNMENT.

35. **NumCondDollarSignWord(block_str)**—If the Dollar Sign Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to NUM_INPUT. Otherwise this function returns the type CONSOLE_MSG.
36. **NumCondSWord(block_str)**—If the S Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to SIN. Otherwise this function returns the type COMMAND.
37. **NumCondCWord(block_str)**—If the C Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to COS. Otherwise this function returns the type COMMAND.
38. **NumCondTWord(block_str)**—If the T Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to FIX. Otherwise this function returns the type COMMAND.
39. **NumCondAWord(block_str)**—If the A Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to ATAN. Otherwise this function returns the type COMMAND.
40. **NumCondRWord(block_str)**—If the R Word comes after a '=', and the character 'L' or 'E' is present, then this function returns the type FUNCTION, and sets the word to SQRT. Otherwise this function returns the type COMMAND.
41. **NumCondEWord(block_str)**—If the E Word comes after a number, or if it begins the block , then this function returns the type VARIABLE_TAG. Otherwise this function returns the type FUNCTIONS, and sets the word to E_FUNC.
42. **OkumaCondNWord(block_str)**—If the next non-space character is numeric and the string "IF" or "GOTO" existed previously on the line, then this routine returns the type CONDITIONAL, and sets the word to GOTO_LABEL. If the next non-space character is numeric and the string "IF" or "GOTO" does not previously exist on the line, then this routine returns the type CONDITIONAL, and sets the word to LABEL. If the next non-space character is not numeric, and the string "IF" or "GOTO" existed previously on the line, then this routine returns the type CONDITIONAL, and sets the word to GOTO_SEQ. Otherwise this routine returns the type COMMAND.
43. **OliCondLeftParent(block_str)**—If the Left Parenthesis Word comes after a "=" character, then this function returns the type LPAREN. Otherwise this function returns the type BEGIN_TYPEII.
44. **OliCondRightParent(block_str)**—If the RightParenthesis Word comes after a BEGIN_TYPEII word, then this function returns the type END_TYPEII. Otherwise this function returns the type RPAREN.
45. **Round(value)**—Returns the integral value closes to the specified value..
46. **Siemens3RCond(block_str)**—Same as SiemensRCond(block_str), except it also looks to see if the @ command had finish prior to this point in the block. It does this by looking backwards from the current point, and checking to see if a non-operator, non-R, non-digit character exists between the current point and the @ command. When the R word is used as a command, it typically calls the VariableArgument macro. This sets up the addition arguments that are then processed by the specific Siemen's macro.
47. **SiemensCondRParenWord()**—If MCALL is on the line, the word type is set to Right Precedence, otherwise it is set to End Comment.
48. **SiemensCondLParenWord()**—If MCALL is on the line, the word type is set to Left Precedence, otherwise it is set to Begin Comment.

- 49. **SiemensRCond(block_str)**—If the R Word comes after a “@” character, then this function returns the type COMMAND. Otherwise this function returns the type VARIABLE_TAG.
- 50. **sinh_d(value)**—Returns the hyperbolic cosine of the given angle specified in degrees.
- 51. **sinh_r(value)**—Returns the hyperbolic cosine of the given angle specified in radians.
- 52. **sin_d(value)**—Returns the cosine of the given angle specified in degrees.
- 53. **sin_r(value)**—Returns the cosine of the given angle specified in radians.
- 54. **sqrt(value)**—Returns the square root of the specified value.
- 55. **tan_d(angle)**—Returns the tangent of the given angle specified in degrees.
- 56. **tan_r(angle)**—Returns the tangent of the given angle specified in radians.
- 57. **tanh_d(angle)**—Returns the hyperbolic tangent of the given angle specified in degrees.
- 58. **tanh_r(angle)**—Returns the hyperbolic tangent of the given angle specified in radians.

Command Line Options & Batch Processing

Command line options for VERICUT

Command line options configure and control VERICUT upon execution. Command line options are particularly useful in customizing the user interface for specific site use, or during batch processing as described in the section later in this chapter titled "Batch processing".

Available command line options are listed below in alphabetical order. See the appropriate section for details. Options can be arranged on the command line in any order.

Command line option:	General description:
<uservariablename>=<uservalue>	Sets a user variable (value in the user interface)
addusr=<addusrfile>	Loads an Add-user file
append_log	Open Log file in append mode
auto-diff=<reportfilename>	Perform an AUTO-DIFF model comparison and write results to a file
batch	Runs VERICUT in the background, without opening the user interface
batchp	Similar to batch, except "pops open" the VERICUT window when finished
datalink	Opens the OptiPath "DataLink" to a software application
export	Exports the VERICUT cut model
fitall	Fits all open views
IMRCAPDT=on	Appends recorded images to an existing Image file
ip=<ipfile>	Loads an In Process file
LGFILNAM=<logfile name>	Set the Log file name
optimize	Turns OptiPath on
record	Records VERICUT images to an Image file
reset_cutcolor	Resets cut colors of all machine cuts
reset_log	Reset the Log file
results	Creates a Results file at then end of cutting
reverse	Reverses the active view
usr=<usrfile>	Loads a User file

Help on VERICUT

zoom_out[=<value>]	Zooms out approximately 20% for each "zoom_out"

Note: If the option you are looking is not listed above, see "Obsolete command line options" at the end of this section.

Windows note: Command line options containing an assignment ("=") must be surrounded with double quotes, for example: vericut "usr=part.usr"

<uservariablename>=<uservalue>

Specifies a user variable and corresponding value to be set in the user interface. Also known as a "user variable override", user variables and values entered on the command line are not case sensitive. However, some operating systems use case sensitivity when reading directory paths and file names. Consult your operating system guide to determine if case sensitivity applies.

Windows example:

```
vericut "FASTFEED=500"
```

UNIX example:

```
vericut FASTFEED=500
```

When combined with other command line options that affect user values, the options are processed in the order listed below. Processing order is the same regardless of the order in which they are listed on the command line.

Order in which command line options affecting user values are processed:

1. ip=<ipfile> - load VERICUT model
2. usr=<usrfile> - load complete set of user values
3. addusr=<addusrfile> - load sub-set of user values
4. <uservariablename>=<uservalue> - load individual user values

Notes:

Specific spelling and formats apply. To see a list of VERICUT user variable names/values, view any User file saved from the current version of VERICUT. The middle column lists user variables, while the right column indicates the saved user values for corresponding user variables.

When specifying multiple word user values join the words with an underscore ("_").

When user values corresponding to user variables in a list format (e.g. MODELLST, TPLIST, VIEWDEFS, etc.), use the addusr command line option (see above) with an Add-user file containing the variable list.

Back to command line options list

addusr=<addusrfile>

Loads an existing Add-user file into VERICUT. The full directory path can be included with the file name.

Help on VERICUT

When combined with other command line options that affect user values, the options are processed in the order listed below. Processing order is the same regardless of the order in which they are listed on the command line.

Order in which command line options affecting user values are processed:

1. ip=<ipfile> - load VERICUT model
2. usr=<usrfile> - load complete set of user values
3. addusr=<addusrfile> - load sub-set of user values
4. <uservariablename>=<uservalue> - load individual user values

[Back to command line options list](#)

append_log

When VERICUT is run and an existing Log file is detected, the Log file is opened in "append" mode. Additional log messages are appear at the end of the file.

See also: reset_log and LGFILNAM=<logfilename>.

[Back to command line options list](#)

auto-diff=<reportfilename>

Performs an AUTO-DIFF model comparison and writes results to the specified AUTO-DIFF Report file. The comparison method and tolerances used are controlled via the **Analysis menu > AUTO-DIFF** window. The comparison is performed when tool path processing has finished. Immediately afterwards, VERICUT automatically restores the model to its "as machined" state, as it was prior to differencing.

This option is commonly used during batch processing to automatically inspect the part after machining.

[Back to command line options list](#)

batch

Runs VERICUT in the background, without opening the user interface. In this mode, VERICUT loads specified IP, User, and Add-user files (in listed order), processes the tool path file(s), and exits-without user interaction. The "batch" option can appear anywhere on the command line. Examples follow.

Windows examples:

```
vericut batch "usr=part.usr"
```

```
vericut "usr=C:\ncparts\part.usr" batch
```

UNIX examples:

```
vericut batch usr=part.usr
```

```
vericut usr=/ncparts/part.usr batch
```

For step-by-step examples of how to accomplish popular tasks via batch processing, see "Batch processing" later in this chapter.

Help on VERICUT

Notes:

The same CGTech security and computer resources are required to run VERICUT in batch mode as required during interactive processing.

Logging out after initiating a batch session has no affect on the process, as long as the computer remains active.

Another form of batch processing, batchp, "pops open" the VERICUT window when finished to reveal the cut workpiece.

A "Batch Wizard" provides a guided, graphical method of creating and running CGTech products in batch mode. See "Batch processing" later in this chapter for details.

[Back to command line options list](#)

batchp

Similar to batch, except when finished the VERICUT window "pops open" to reveal the cut workpiece. The model can then be analyzed or saved for later use.

See "Batch processing" later in this chapter for details.

datalink

Opens the OptiPath "DataLink" and begins passing material removal data to the software application specified in the **OptiPath menu > DataLink** window.

[Back to command line options list](#)

export

Exports the VERICUT cut model. The method and tolerances used are controlled via the **File menu > Export Model** window configuration. The export operation is performed when tool path processing has finished.

[Back to command line options list](#)

fitall

Fits all open views. Use this option to ensure models of the workpiece, holding fixtures, and NC machine are fit completely within all open views.

[Back to command line options list](#)

IMRCAPDT=on

Actually a user variable override, this option is used with the "record" option to append images recorded via the **File menu > Images > Record** function to an existing Image file.

[Back to command line options list](#)

ip=<ipfile>

Help on VERICUT

Loads an existing In-process file into VERICUT. The full directory path can be included with the file name. The examples that follow assume an In-process file named “part.ip” has been saved.

Windows examples:

```
vericut “ip=part.ip”
```

```
vericut “ip=C:\ncparts\part.ip”
```

UNIX examples:

```
vericut ip=part.ip
```

```
vericut ip=/ncparts/part.ip
```

When combined with other command line options that affect user values, the options are processed in the order listed below. Processing order is the same regardless of the order in which they are listed on the command line.

Order in which command line options affecting user values are processed:

1. ip=<ipfile> - load VERICUT model
2. usr=<usrfile> - load complete set of user values
3. addusr=<addusrfile> - load sub-set of user values
4. <uservariablename>=<uservalue> - load individual user values

Back to command line options list

LGFILNAM=<logfile name>

Actually a user variable override, this option sets the name of the Log file to receive error, warning, and informational messages resulting from processing a tool path in VERICUT.

See also: append_log and reset_log.

Back to command line options list

optimize

Turns OptiPath on for optimizing tool path feed rates and/or spindle speeds. Cutting tools optimized and optimization values used are controlled via the **Setup menu > Toolpath** and **OptiPath menu > Control** window configurations.

Back to command line options list

record

Records VERICUT session images to an Image file, as specified by the **File menu > Images > Record** window configuration. When record is used with the IMRCAPDT=on option, images are appended to the end of the file.

Back to command line options list

Help on VERICUT

reset_cutcolor

Resets the cut colors of all machine cuts to be one color: the Reset Cut color. Cut colors are reset prior to cutting.

Back to command line options list

reset_log

When VERICUT is run and an existing Log file is detected, the Log file is reset (cleared) prior to writing new log messages.

See also: `append_log` and `LGFILNAM=<logfilename>`.

Back to command line options list

results

Creates a Results file at then end of cutting. This HTML file can be viewed interactively in VERICUT via clicking **Info menu > Results File**, or via your HTML browser.

Back to command line options list

reverse

Reverse the active view prior to cutting. This option is useful when recording images and tool motion occurs on the opposite side of the model.

See "**Reverse**" under **View menu > Orient** in the *Reference* chapter for more information.

Back to command line options list

usr=<usrfile>

Loads an existing User file into VERICUT. The full directory path can be included with the file name. The examples that follow assume a User file named "part.usr" has been saved.

Windows examples:

```
vericut "usr=part.usr"
```

```
vericut "usr=C:\ncparts\part.usr"
```

UNIX examples:

```
vericut usr=part.usr
```

```
vericut usr=/ncparts/part.usr
```

When combined with other command line options that affect user values, the options are processed in the order listed below. Processing order is the same regardless of the order in which they are listed on the command line.

Order in which command line options affecting user values are processed:

Help on VERICUT

1. ip=<ipfile> - load VERICUT model
2. usr=<usrfile> - load complete set of user values
3. addusr=<addusrfile> - load sub-set of user values
4. <uservariablename>=<uservalue> - load individual user values

Back to command line options list

zoom_out[=<value>]

Zooms out, or decreases the viewing size of uncut stock models by approximately 20% for each “**zoom_out**” listed. For example, “**zoom_out**” zooms out one step, while “**zoom_out=2**” zooms out twice.

See “**Zoom Out**” under **View menu > Orient** in the *Reference* chapter for more information.

Back to command line options list

Obsolete command line options

The command line options listed below are obsolete in this version of VERICUT. Recommended alternate command line options are recommended where possible.

autodemo=<imgfile>—Not supported.

batcht—“True batch”, obsolete- use “batch” instead.

OpenGL—Not supported.

pbappend—Obsolete- use “IMRCAPDT=on” instead.

pseudocolor—No longer applies.

slice—Not supported in this release.

statdisp=<computername>—Not supported.

truecolor—No longer applies.

Back to command line options list

Command line options for CGTech converters

CGTech converter products provide a way to translate a wide variety of CAD/CAM data, primarily for use in VERICUT. All converters can be run interactively, or in batch mode. When run in batch, command line options are used to configure the converters without the use of the user interface.

The sections that follow describe command line options available for CGTech converter products. See the appropriate section for details.

Binary CL Converter command line options

IGES Converter command line options

Mastercam NCI Converter command line options

PolyFix Converter command line options

Surface to Solid Converter command line options

VDA Converter command line options

Windows note: Command line options containing an assignment (“=”) must be surrounded with double quotes, for example: iges2v “input=inputfile.igs”

Binary CL Converter command line options

Command line options available for the Binary CL Converter are described in the table below.

Option	Description	Default
batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	Binary CL Input File	
output=output-file	Output File (ASCII APT)	
table=table-file	APT Table File	aptwords.tbl (CGTech library)
cltype={1 2 3 4}	Input File Type: 1=IBM,2=IBMWS,3=CATWS,4=VAXWS	1
circles={YES NO}	Output Circles	Yes
height=value	Minimum Cutter Height	0

IGES Converter command line options

Help on VERICUT

Command line options available for the IGES Converter are described in the table below.

Option	Description	Default
batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	IGES Input File	
output=output-file	Output File	
dtool=value	Deviation Tolerance	0.05 for in. model, 1.27 for mm model
format={ASCII BINARY}	Output File Format	BINARY
type={VERICUT STL}	Output File Type	STL
mode={TRIMMED BASE ALL}	Surface Input Mode	ALL
dir={INWARD OUTWARD GUESS}	Normal Direction	OUTWARD
summary	Summary of input file contents	

Mastercam NCI Converter command line options

Command line options available for the Mastercam NCI Converter are described in the table below.

Option	Description	Default
batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	NCI Input File	
output=output-file	Output File (ASCII APT)	
data={TOOLCENTER TOOL TIP}	Control Point	Tool Tip
circle_tol=value	Circle Tolerance	
tool_change_height=Zvalue	Tool Change Retract Height	
cutter_height=value	Cutter Height	

PolyFix Converter command line options

Command line options available for the IGES Converter are described in the table below.

Option	Description	Default
batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	Input File	
intype={ VERICUT STL}	Input File Type	STL
output=output-file	Output File	
outformat={ ASCII BINARY}	Output File Format	BINARY
outtype={ VERICUT STL}	Output File Type	STL

Surface to Solid Converter command line options

Command line options available for the Surface to Solid Converter are described in the table below.

Option	Description	Default
batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	IGES Input File	
output=output-file	Output File	
tol=value	Tolerance	
intype={ VERICUT STL}	Input File Type	
outtype={ VERICUT STL}	Output File Type	
format={ ASCII BINARY}	format={ ASCII BINARY}	BINARY
matrix=matrix-string	Transformation Matrix	Identity matrix
offset=value	surface Offset	0
base=value	Base Location	0
rectangle={ NO YES}	Project Rectangle	NO

VDA Converter command line options

Command line options available for the VDA Converter are described in the table below.

Option	Description	Default

Help on VERICUT

batch	Execute without user interface	
batcht (obsolete)	Same as "batch"	
input=input-file	VDA Input File	
output=output-file	Output File	
dtol=value	Deviation Tolerance	0.05 for in. model, 1.27 for mm model
format={ASCII BINARY}	Output File Format	BINARY
type={VERICUT STL POINT}	Output File Type	STL
dir={INWARD OUTWARD GUESS}	Normal Direction	OUTWARD

Batch processing

This topic is still being written.

VERICUT Batch Wizard

Environment variables

You can set environment variables to help customize the VERICUT user interface, alter VERICUT performance, and perform other various actions. This section describes environment variable formats and the available environment variables that can affect the VERICUT simulation.

UNIX versus Windows formats

Environment variables are typically set via CGTech command files located in the CGTech "commands" directory, under the directory representing your computer type. For example, the command files for VERICUT 5.0 installed on the "C:" drive of a Windows NT computer are located in the C:\CGTech50\win_nt\commands directory. Environment variables can also be set in any other system file used to execute VERICUT.

Formats used to set environment variables differ, based on the computer's operating system. Consult the operating system documentation, or refer to the CGTech-installed command files on your computer for specific examples.

Windows general format:

```
set ENV_VAR_NAME=value
```

Windows examples:

```
set CGTECH_DEFAULT_UNITS=inch  
set CGTECH_LIBRARY=C:\CGTech50\library
```

UNIX general format (2 records required to set each variable):

```
ENV_VAR_NAME=value  
export ENV_VAR_NAME
```

UNIX examples:

```
CGTECH_DEFAULT_UNITS=inch  
export CGTECH_DEFAULT_UNITS  
CGTECH_LIBRARY=C:\CGTech50\library  
export CGTECH_LIBRARY
```

CGTech environment variables

Available environment variables that can affect the VERICUT simulation are listed below in alphabetical order. For simplicity, all examples of use below are in Windows format. See above for UNIX format examples.

CGTECH_DEFAULT_UNITS—Sets the default unit measurement system (inch or millimeter) for pre-4.0 User files that do not contain a "UNITS" value.

Help on VERICUT

Example of use: set CGTECH_DEFAULT_UNITS=millimeter

CGTECH_FONT_SIZE—Sets the the font size for text appearing in the VERICUT user interface. Size can be given an absolute point size, or incremental value from the default value of "12".

Examples of use:

- set CGTECH_DEFAULT_UNITS=14 (sets 14 pt. font)

- set CGTECH_DEFAULT_UNITS=+2 (also sets 14 pt. font: default 12 +2= 14)

CGTECH_STATUS—Creates a Status file, an ASCII text file that contains whatever data is sent to the Status window during the simulation. The full directory path can be included with the file name.

Example of use: set CGTECH_STATUS=C:\mydir\statusfile.txt

CGTECH_USERNAME—Specifies the user name to include in the name of the Preferences file that is automatically saved when you exit VERICUT. This environment variable is recommended to prevent users that share a machine (or an account) from overwriting the Preferences file.

Example of use: set CGTECH_USERNAME=bob (Preferences file saved => "cgttech_bob.prefs")

CGTECH_VC_IP_FILTER—Sets the default filter in file selection windows for opening and saving In-process files.

Example of use: set CGTECH_VC_IP_FILTER=C:\mydir*.ip

CGTECH_VC_USR_FILTER—Sets the default filter in file selection windows for opening and saving User files.

Example of use: set CGTECH_VC_USR_FILTER=C:\mydir*.usr

CGTECH_VCUSR—Specifies the default User file for VERICUT. The full directory path can be included.

Example of use: set CGTECH_VCUSR=C:\mydir\myuserfile.usr

Adding custom environment variables for shortcuts

You can add your own environment variables to the beginning section of the VERICUT command file and have them appear in the shortcut list of file selection windows. VERICUT lists all custom environment variables that begin with "CGTECH_" in the shortcut list.

For example, the following environment variable creates a shortcut to V4.4 VERICUT sample files:

set CGTECH_44SAMPLES=C:\CGTech44\samples\vericut

Chapter 4 VERICUT Files

File Types and Descriptions

The sections that follow provide information about the files VERICUT uses, such as: typical file extensions, file contents and formats, and how to open and save them. (See list of VERICUT file types and extensions)

Types of VERICUT files

Files provided by CGTech to assist with simulating tool path and machine tool motions can be divided into two categories:

- Library files**—General purpose files for configuring VERICUT to simulate tool path and machine motions. Library files are included in every installation, and include:
 - VERICUT default files—Default files opened by VERICUT. The files are configured to perform a short simulation-just press the Play button (green arrow pointing right). Two sets of files provide support for inch (vericut.*) and metric (vericutm.*) environments.
 - VERICUT initialization files—Initialization files opened by VERICUT via **File menu > New**. Two sets of files provide support for inch (init.*) and metric (initm.*) environments.
 - "Master" Machine and Control files—Pre-configured NC machines and controls intended to help you configure VERICUT for simulating G-code tool paths. See "Machine files" and "Control files" later in this chapter for details.
- Sample files**—Files used to demonstrate VERICUT capabilities and options. Sample files are included with CGTech software on the CD, and are available to all users who choose to load them. For information about these files, see "Sample-demo files" later in this chapter.

Files and descriptions

The types of files used by VERICUT to load and save various types of data are listed below in alphabetical order. See the appropriate section for a description of each.

File Type	Typical Extension
Add-user file	.usr
APT Output file	.tp
APT Table file	.tbl
AUTO-DIFF Report file	.rpt
CGTech Macro Executable file (CME file)	.cme
Control file	.ctl

Help on VERICUT

Control Report file	.rpt
Design Points file	.pts
G-Code Log file	.log
G-Code Report file	.rpt
IGES model file	.igs
Image file	.img
In Process file	.ip
Job file	.job
Log file (VERICUT log file)	.log
Machine file	.mch
Optimized file (optimized tool path file)	.mco (G-code), .tpo (APT)
OptiPath Library file	.olb
Preferences file	.prefs ("cgtech_user.prefs")
Results file	.htm (HTML)
Status file	.txt
Stereolithography model file	.stl
Tool Library file	.tls
Toolpath file	.mcd (G-code), .tp (APT), many other extensions depending on CAM system
User file	.usr
VERICUT Linear-sweep file	.swp
VERICUT Polygon file	.dsn (design), .fix (fixture), .stk (stock), .ply (any)
VERICUT Solid-of-revolution file	.sor

Add-user file

Typical file extension: .usr

An ASCII text file that contains a sub-set of User file records. This file is typically created via text editor or custom routine.

Add-user files are loaded into VERICUT via the **File menu > Open** function. Add-user files can also be specified on the command line when VERICUT is executed, such as during batch processing to simulate tool paths without user interaction.

Help on VERICUT

An Add-user file must contain the following records, in listed order:

VERICUT-user

Version-x.x

<--- User variables/values go here

END_USER_FILE

where "x.x" is the current software version. User values to be set by the Add-user file follow the "Version" line, but come before the "END_USER_FILE" record which designates the end of the file. Spaces are not significant.

Note: User values defined in the "list" format, e.g.: MODELST, VIEWDEFS, VIEWLIST, TPLIST, etc. must include the entire list, including "BEGIN..." and "END..." entries.

Hint: To see the proper format for user values you wish to save in an Add-user file, first save them in a User file, then cut and paste the desired entries to avoid typing or format errors that will invalidate the file.

Back to file types list

APT Output file

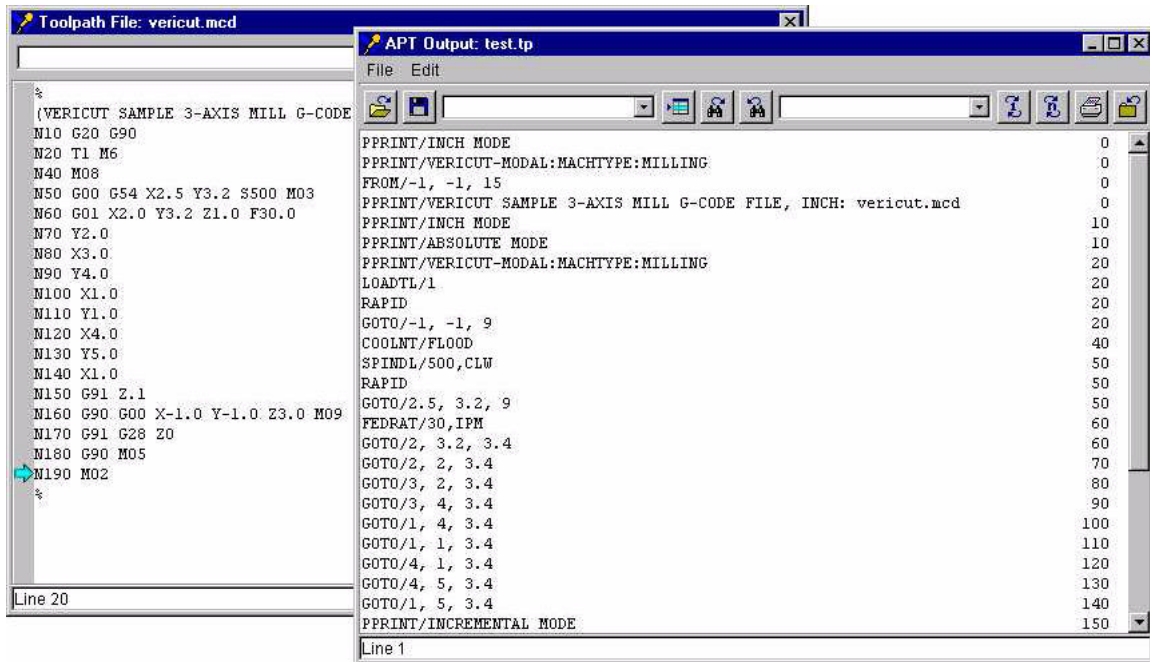
Typical file extension: .tp

An ASCII text file containing APT tool path records resulting from reverse post-processing by VERICUT. By default when sequenced G-code data is converted, columns 73-80 in the output file indicate the block numbers of the G-code data block where the converted records came from. Sequence number output is controlled via the "**Output Block Sequence Numbers**" feature on the **Setup menu > G-Code > Process Options > Messages** window.

An APT Output file is created when a G-code tool path is processed with **Create APT Output File** active. (Ref. **Setup menu > G-Code > Process Options > Output Files**) You can view this file via **Setup menu > G-Code > Process Options > View APT Output File**.

Sample APT Output file:

Help on VERICUT



Back to file types list

APT Table file

Typical file extension: .tbl

An ASCII text file used by the Binary CL Converter that contains information necessary to convert binary APT data into ASCII APT tool path records. Such information includes: blocking factor, major and minor word classes, sub-classes, etc.

APT Table files are loaded into VERICUT via the **File menu > Convert > Binary CL** function. APT Table files can also be specified on the command line when the Binary CL Converter is executed, such as during batch processing to simulate tool paths without user interaction.

Sample APT Table file "aptwords.tbl" (in CGTech library):


```

C:\CGTech44\win_nt\bincl2v\aptwords.tbl
File Edit

IBMCL2V-apt
version-2.0

This file contains all of the user configurable information
neccessary for the APT/AC converter to successfully convert
a binary APT CL file to a man-readable APT CL file.

The order of information in this file is important, and should be
as follows:

    Binary CL file BLOCKING FACTOR (via BLOCKING_FACTOR = number)
    Major words table
    Minor words table

The following record specifies the IBM binary CL file Blocking Factor:

BLOCKING_FACTOR = 3228

This table represents all of the APT Major words that the
APT/AC converter will process.

MAJOR_WORDS_BEGIN
$$      WORD          CLASS      SUBCLASS
$$-----
      DRAWLI          2000        17
      DRESS           2000         8
      END             2000         1
      FACEML          2000        22
      GOCLER          2000        16
Line 10
  
```

Tips:

The default blocking factor for interpreting binary CL data is 3228. If this does not produce favorable results, use any text editor to change the blocking factor. Other typical blocking factors you can try are: 512, 4628, 15476.

If the ASCII tool path file output by the converter has major or minor words other than what you expected, try editing the APT Table file to change the "Word" output for a specific APT class/subclass.

Back to file types list

AUTO-DIFF Report file

Typical file extension: .rpt

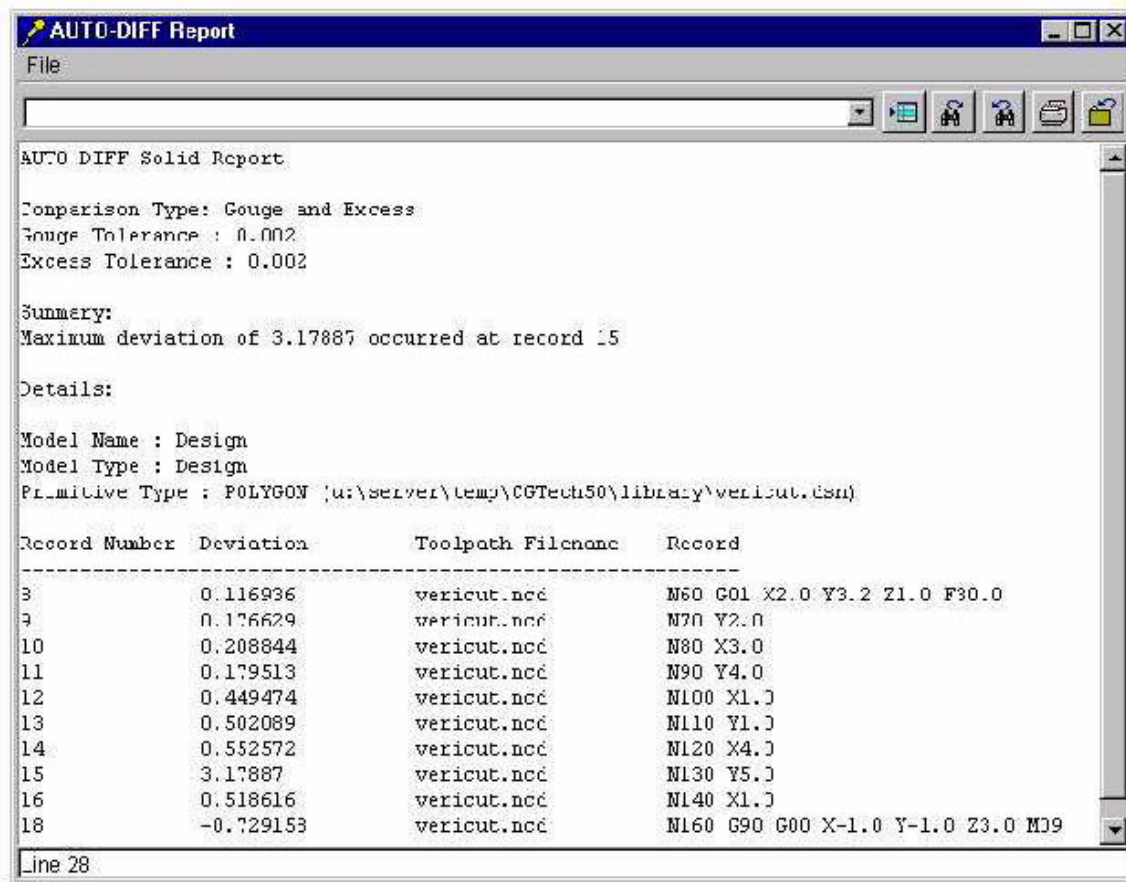
An ASCII text file that contains information about an AUTO-DIFF model comparison operation, such as identification of all detected errors, and the tool path history related to each error. This information is valuable when determining the correctness of the VERICUT simulated cut model, and tool path records responsible for discrepancies detected by AUTO-DIFF.

Help on VERICUT

An AUTO-DIFF Report file is created when an AUTO-DIFF comparison is performed, and can be viewed via the "**View Rpt**" feature. (Ref. **Analysis menu > AUTO-DIFF**)

The exact format of the report varies, but generally includes a summary of the results, information about tolerances and models used in the comparison, error types (gouge/excess) and amounts, tool path history associated with errors, etc.

Sample AUTO-DIFF Report file:



Back to file types list

CGTech Macro Executable file

Typical file extension: .cme

A platform-independent binary file also known as a "CME file", that is compiled using the Machine Developer's Kit and contains custom macros that control processing G-code data. In addition to macros, this file can also establish modals which allow users to further customize how machine code data is interpreted, and provide access to custom tables containing job and machine related data. The CME file used to perform these tasks is specified via the Control file- ref. "**Dev Kit CME**" under the **Setup menu > Control > Adv. Options** function in the *Reference* chapter.

Back to file types list

Control file

Help on VERICUT

Typical file extension: .ctl

An ASCII text file that contains data describing how the NC control processes G-codes. Data in this file includes: a list of interpretable machine codes, machine code format information, rules for grouping machine codes, and rules for calling macros. This file is required to interpret G-code data via the G-Code Data tool path type. When the Machine Developer's Kit is used to create a custom CME file, the Control file can also specify a custom CME file to assist with the simulation.

By default, VERICUT is configured with a "Fanuc" control. Many control configurations are available in the library that is installed with VERICUT software- see "Control files" in the *VERICUT Files* chapter for more information.

A different Control file can be loaded into VERICUT via the **Setup menu > Control > Open** function. NC control configurations are changed via other "Control" functions in the **Setup menu**. Save the new file via **Setup menu > Control > Save As**.

Back to file types list

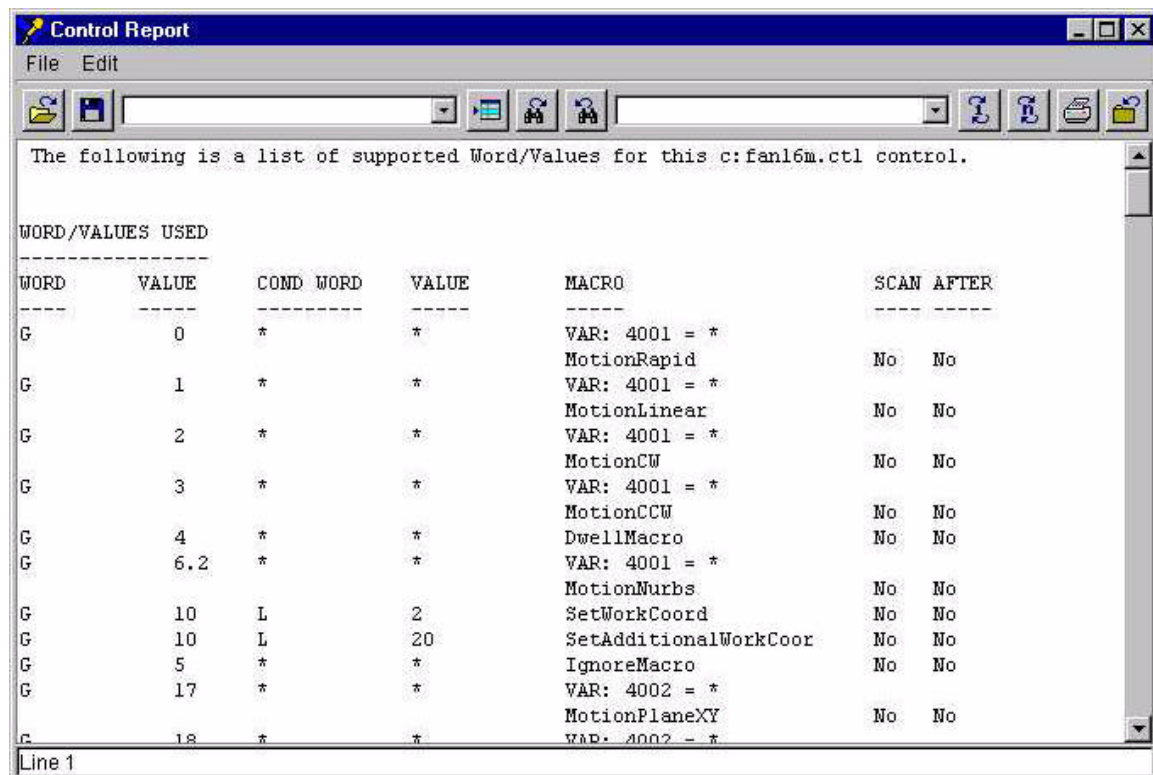
Control Report file

Typical file extension: .rpt

An ASCII text file that contains information about about how the current NC control configuration will interpret various codes. Reported information includes:

G-codes present in the tool path and macros (actions) the control will perform, variables used (if any), cutting tools used, subroutines defined and referenced, and more. Control Report files are created via the **Info menu > Control Report** function.

Sample Control Report file:



The following is a list of supported Word/Values for this c:\fan16m.ctl control.

WORD	VALUE	COND	WORD	VALUE	MACRO	SCAN	AFTER
G	0	*		*	VAR: 4001 = *	No	No
G	1	*		*	MotionRapid	No	No
G	2	*		*	VAR: 4001 = *	No	No
G	3	*		*	MotionLinear	No	No
G	4	*		*	VAR: 4001 = *	No	No
G	6.2	*		*	MotionCW	No	No
G	10	L	2		VAR: 4001 = *	No	No
G	10	L	20		MotionCCW	No	No
G	5	*		*	DwellMacro	No	No
G	17	*		*	VAR: 4001 = *	No	No
G	18	*		*	MotionNurbs	No	No
					SetWorkCoord	No	No
					SetAdditionalWorkCoor	No	No
					IgnoreMacro	No	No
					VAR: 4002 = *	No	No
					MotionPlaneXY	No	No
					VAR: 4002 = *		

Line 1

Help on VERICUT

Back to file types list

Design Points file

Typical file extension: .pts

An ASCII text file also known as an "inspection point file", that contains point locations used to inspect the VERICUT model when performing an AUTO-DIFF Point comparison operation.

Design Point files are loaded into VERICUT via the **Model menu > Model Definition** function. Point record formats support specifying surface normals, tolerances, and check distances with any or all inspection points. Example point record formats follow.

Design Points file record formats:

GOTO/X,Y,Z -where "X,Y,Z" represents the design point location, for example: "GOTO/1,2,3".

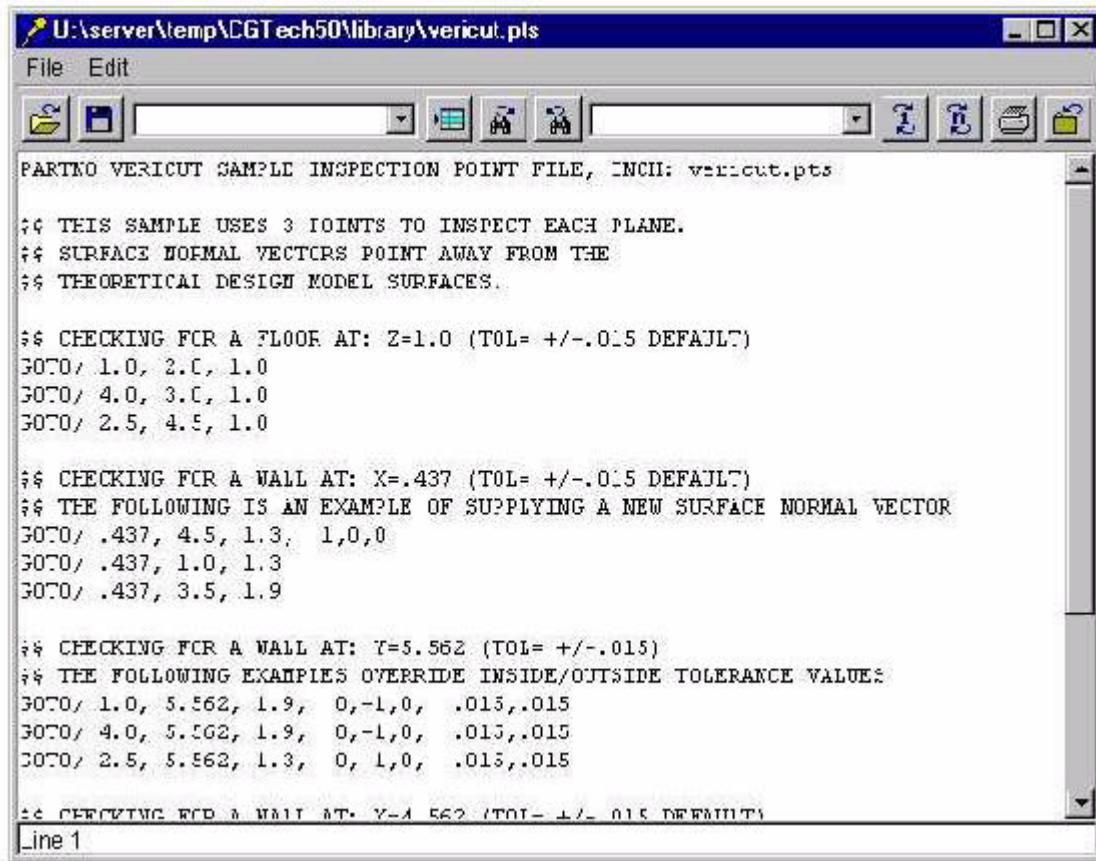
GOTO/X,Y,Z,I,J,K -as above plus "I,J,K" represents the surface normal vector, for example: "GOTO/1,2,3,1,0,0".

GOTO/X,Y,Z,I,J,K,intol,outtol -as above plus "intol" and "outtol" override the Gouge Tolerance and Excess Tolerance values, for example: "GOTO/1,2,3,1,0,0,.0005,.015". The overrides are only applied to the point where they are listed.

GOTO/X,Y,Z,I,J,K,intol,outtol,incheck,outcheck -as above plus "incheck" and "outcheck" override the Gouge Check Distance and Excess Check Distance values, for example: "GOTO/1,2,3,1,0,0,.015,.015,.300,.300". The overrides are only applied to the point where they are listed.

Any combination of the above formats can be included in the Design Points file. There is no limit to number of records that the file can contain. "\$\$" comment records and spaces are permitted.

Sample Design Points file "vericut.pts" (in CGTech library):

A screenshot of a Windows-style text editor window titled "U:\server\temp\CGTech50\library\vericut.pts". The window has a menu bar with "File" and "Edit", and a toolbar with icons for file operations and editing. The text content is as follows:

```
PARTNO VERICUT SAMPLE INSPECTION POINT FILE, INCH: vericut.pts

%% THIS SAMPLE USES 3 POINTS TO INSPECT EACH PLANE.
%% SURFACE NORMAL VECTORS POINT AWAY FROM THE
%% THEORETICAL DESIGN MODEL SURFACES.

%% CHECKING FOR A FLOOR AT: Z=1.0 (TOL= +/-0.015 DEFAULT)
3070/ 1.0, 2.0, 1.0
3070/ 4.0, 3.0, 1.0
3070/ 2.5, 4.5, 1.0

%% CHECKING FOR A WALL AT: X=.437 (TOL= +/-0.015 DEFAULT)
%% THE FOLLOWING IS AN EXAMPLE OF SUPPLYING A NEW SURFACE NORMAL VECTOR
3070/ .437, 4.5, 1.3, 1,0,0
3070/ .437, 1.0, 1.3
3070/ .437, 3.5, 1.9

%% CHECKING FOR A WALL AT: Y=5.562 (TOL= +/-0.015)
%% THE FOLLOWING EXAMPLES OVERRIDE INSIDE/OUTSIDE TOLERANCE VALUES
3070/ 1.0, 5.562, 1.9, 0,-1,0, .015,.015
3070/ 4.0, 5.562, 1.9, 0,-1,0, .015,.015
3070/ 2.5, 5.562, 1.3, 0, 1,0, .015,.015

%% CHECKING FOR A WALL AT: Y=4.562 (TOL= +/-0.015 DEFAULT)
_line 1
```

[Back to file types list](#)

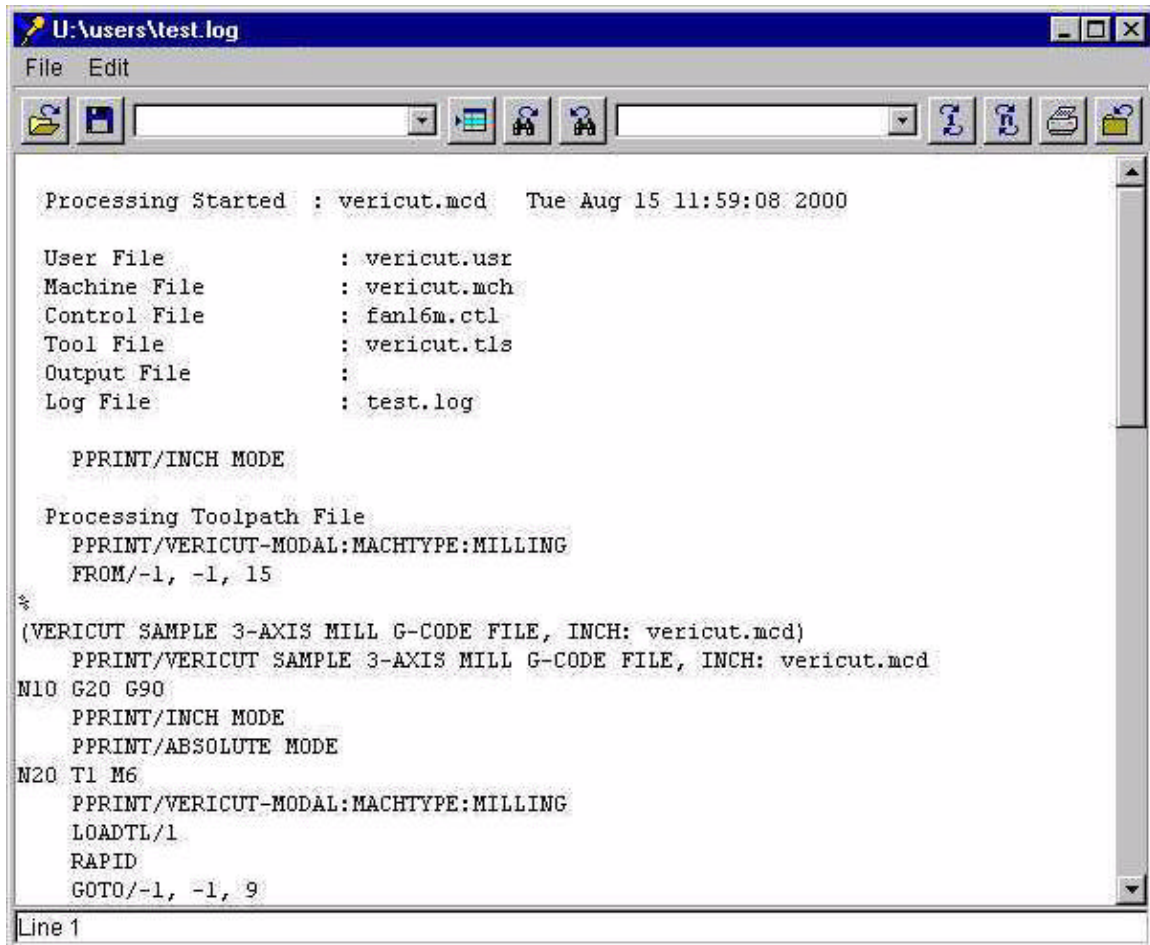
G-Code Log file

Typical file extension: .log

An ASCII text file that contains error, warning and informational messages about G-code processing. One of the most valuable pieces of information this file provides is a copy of each input G-code data block followed by the converted ASCII APT tool path record indented by 4 spaces. This allows you to see a direct translation of G-code versus equivalent ASCII APT output for all tool path records processed.

G-code Log files are created when a name is entered in the **Setup menu > G-Code > Process Options > Output Filenames: Log File** field, and are viewed via **Setup menu > G-Code > Process Options > View G-Code Log**. For help with interpreting error and warning messages in this file, consult the *ERRORs and Problem Solving* chapter.

Sample G-Code Log file:



```
U:\users\test.log
File Edit

Processing Started : vericut.mcd Tue Aug 15 11:59:08 2000

User File      : vericut.usr
Machine File   : vericut.mch
Control File   : fan16m.ctl
Tool File      : vericut.tls
Output File    :
Log File       : test.log

PPRINT/INCH MODE

Processing Toolpath File
PPRINT/VERICUT-MODAL:MACHTYPE:MILLING
FROM/-1, -1, 15
%
(VERICUT SAMPLE 3-AXIS MILL G-CODE FILE, INCH: vericut.mcd)
PPRINT/VERICUT SAMPLE 3-AXIS MILL G-CODE FILE, INCH: vericut.mcd
N10 G20 G90
PPRINT/INCH MODE
PPRINT/ABSOLUTE MODE
N20 T1 M6
PPRINT/VERICUT-MODAL:MACHTYPE:MILLING
LOADTL/1
RAPID
GOTO/-1, -1, 9

Line 1
```

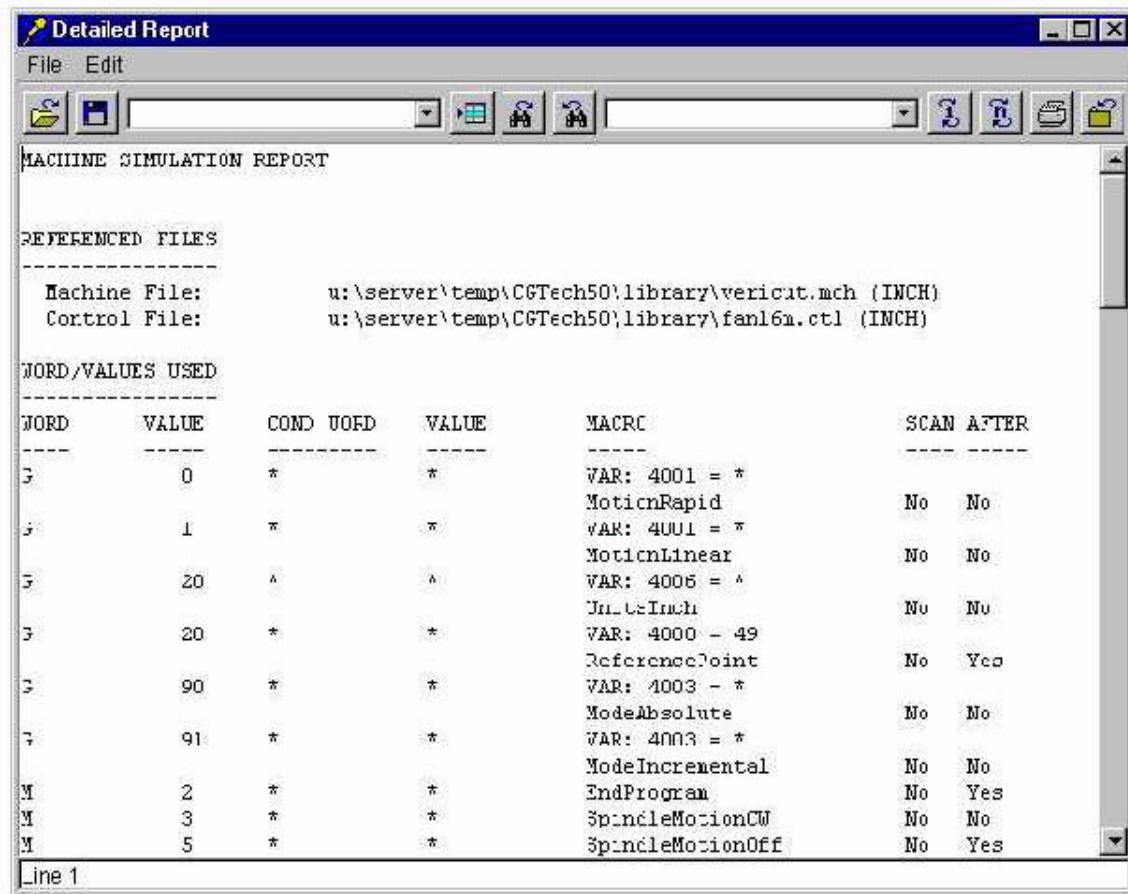
[Back to file types list](#)

G-Code Report file

Typical file extension: .rpt

An ASCII text file that contains information about how the current NC control configuration will interpret a G-code tool path file. Reported information includes: G-codes present in the tool path and macros (actions) the control will perform, variables used (if any), cutting tools used, subroutines defined and referenced, and more. G-code Report files are created via the Info menu > G-code Report function.

Sample G-Code Report file:



[Back to file types list](#)

IGES model file

Typical file extension: .igs

An IGES (Initial Graphics Exchange Specification) model file, also known as an "IGES file", is an ASCII text file which describes virtually anything that can be modeled in a CAD system. IGES data is widely accepted as an industry standard for transferring model data. While IGES model files containing only 2-D data (lines and arcs) are not useful in VERICUT, files containing 3-D surfaces and solids can be used to describe stock, fixture, or design models, as well as machine component shapes.

IGES model files are imported into VERICUT via the **Model menu > Model Definition** function, or can be converted into STL or VERICUT Polygon model files using the IGES converter.

[Back to file types list](#)

Image file

Typical file extension: .img

A platform-independent binary file that contains recorded images of a tool path simulation. Image files can contain any combination of snapshot (single still image) and animation images (similar to a movie). These files can be recorded and replayed on any supported computer platform.

Help on VERICUT

Two types of VERICUT images can be recorded. Both types can be recorded into the same file. Machining status is automatically recorded with either image type.

1. Snapshot—single-frame image. These images are also used to mark stop points in an animation.
2. Animation—multiple images that when replayed, appear like a simulation (like a movie).

Image files are saved (recorded) via the **File menu > Images > Record** function, and replayed via **File menu > Images > Playback**.

Note: Image files are not intended for printing. Use **File menu > Print** or **File menu > Images > ShadeCopy** to print VERICUT images.

Back to file types list

In Process file

Typical file extension: .ip

A part ASCII text-part binary file also known as an "IP file" that contains the data necessary to re-establish a VERICUT session, including the current VERICUT model (with cuts), user interface settings, and a copy of the Log file. IP files are also used to "backup" the verification session, safeguarding against lost work if the session is interrupted by computer or power failure.

Save an In-process file interactively via **File menu > In Process > Save As**, or automatically via the **Auto Save** function. Due to differences in math processing on various computer types, IP files are usable on computers of the same type on which they were saved.

Load IP files via **File menu > In Process > Open** to re-establish a VERICUT session or provide a VERICUT model for additional machining and analysis. IP files can also be specified on the command line when VERICUT is executed, such as during batch processing to simulate tool paths without user interaction.

Note: Environment variables can be used to affect VERICUT's use of IP files. For more information, see "Environment variables" in the *Reference* chapter.

Back to file types list

Job file

Typical file extension: .job

An ASCII text file that contains job data and references to support files required by pre-5.0 versions of VERICUT to directly interpret G-code tool path files.

VERICUT 5.0 or higher no longer uses this file—all user and job-related data is now combined in the User file. Pre-5.0 Job files can be loaded in VERICUT 5.0 via **File>Open**, but only a User file can be saved. To save the Job file information in 5.0 files, be sure to save a User file.

Back to file types list

Log file

Typical file extension: .log

An ASCII text file that contains information about a VERICUT session, such as: names of tool paths

Help on VERICUT

processed, errors, warnings, simulated machining times, and other informational messages.

When VERICUT is run, it must always be able to open a Log file. When an existing Log file of the same name is found VERICUT responds differently, depending on the run mode:

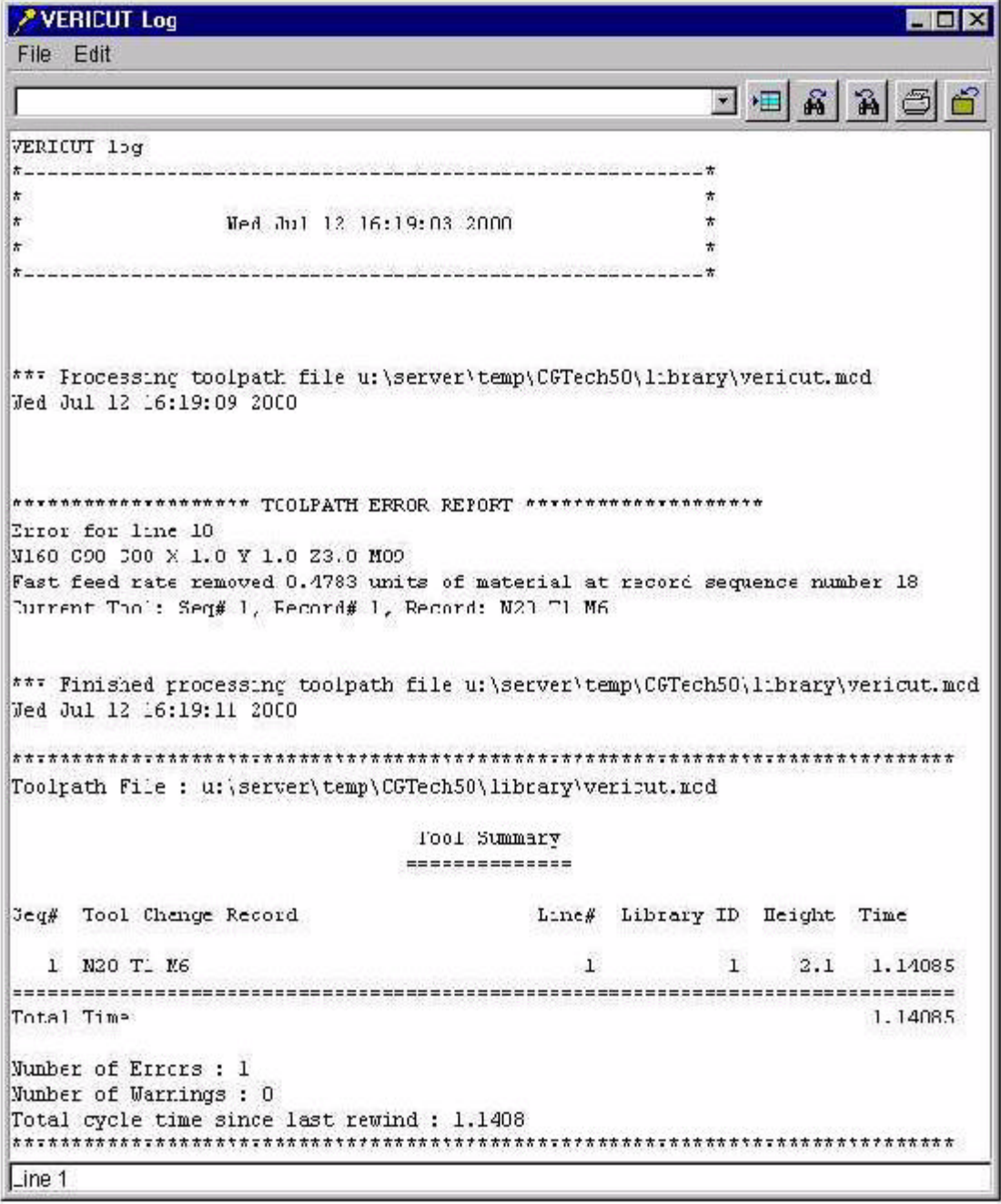
VERICUT run in interactive mode—Log file is automatically reset, unless the "append_log" command line options is used.

VERICUT run in batch mode—Log file is automatically reset, unless the "reset_log" command line options is used.

For more information on command line options used with VERICUT, see "Command line options for VERICUT" in the *Reference* chapter.

Using the **Info menu > VERICUT Log** function, Log file contents can be viewed, edited, or the file name changed. For help with interpreting error and warning messages in this file, consult the *ERRORs and Problem Solving* chapter.

Sample VERICUT Log file:



```

VERICUT Log
File Edit

VERICUT log
*-----*
*
*           Wed Jul 12 16:19:03 2000
*
*-----*

*** Processing toolpath file u:\server\temp\CGTech50\library\vericut.mcd
Wed Jul 12 16:19:09 2000

***** TOOLPATH ERROR REPORT *****
Error for line 10
N160 C90 D00 X 1.0 Y 1.0 Z3.0 M09
Fast feed rate removed 0.4783 units of material at record sequence number 18
Current Tool: Seq# 1, Record# 1, Record: N20 T1 M6

*** Finished processing toolpath file u:\server\temp\CGTech50\library\vericut.mcd
Wed Jul 12 16:19:11 2000

*****
Toolpath File : u:\server\temp\CGTech50\library\vericut.mcd

                        Tool Summary
                        =====

Seq#  Tool Change Record                Line#  Library ID  Height  Time
-----
   1   N20 T1 M6                        1         1       2.1   1.14085
=====
Total Time=                                1.14085

Number of Errors : 1
Number of Warnings : 0
Total cycle time since last rewind : 1.1408
*****
Line 1
  
```

Back to file types list

Machine file

Typical file extension: .mch

An ASCII text file that contains data describing the construction, kinematics, and other properties of an NC machine tool. By default, VERICUT is configured with a 3-axis vertical mill machine. Many machine configurations are available in the library that is installed with VERICUT software- see "Machine files" in the *VERICUT Files* chapter for more information.

Help on VERICUT

A different Machine file can be loaded into VERICUT via the **Setup menu > Machine > Open** function. NC machine configurations are changed via other "**Machine**" functions in the **Setup menu**. Save the new file via **Setup menu > Machine > Save As**.

Back to file types list

Optimized file

Typical file extension: .mco (G-code tool path file) or .tpo (APT tool path file)

The Optimized tool path file is created from optimizing tool path feed rates via OptiPath.

For details about using OptiPath and sample optimized tool path files, see the **OptiPath menu** functions in the *Reference* chapter.

Back to file types list

OptiPath Library file

Typical file extension: .olb

An ASCII text file that contains optimization data for use by VERICUT's OptiPath feed rate optimization module. Specific optimization data values are defined per tool intended to be used in specific machining conditions, e.g. workpiece material, NC machine, etc.

This file is created and maintained via the **OptiPath menu > Manager** function. The OptiPath Library file to use during tool path optimization is specified via **OptiPath menu > Control** and is used by VERICUT when optimizing tool path feed rates via turning OptiPath on in the same window.

Tool path optimization can be performed in a variety of ways, including: configuring cutting tools-to-optimization library records, storing optimization information with tools in a VERICUT Tool Library, or interactive prompting.

Back to file types list

Preferences file

Typical file extension: .prefs

Also known as a "Prefs file", this ASCII text file is automatically saved when you exit VERICUT and stores such user preferences as:

- VERICUT user interface Look & Feel

- Size and locations of VERICUT windows

- State of menu toggles in any window that has them, for example: **View menu > Toolbar, Axes**, etc.

- Filters for VERICUT file selection windows

- Recent User files opened in VERICUT

The Preferences file is unique for each user, based on your login. The file is named "cgtech_user.prefs" and is automatically stored in your home directory. User preferences are updated each time you exit VERICUT.

Help on VERICUT

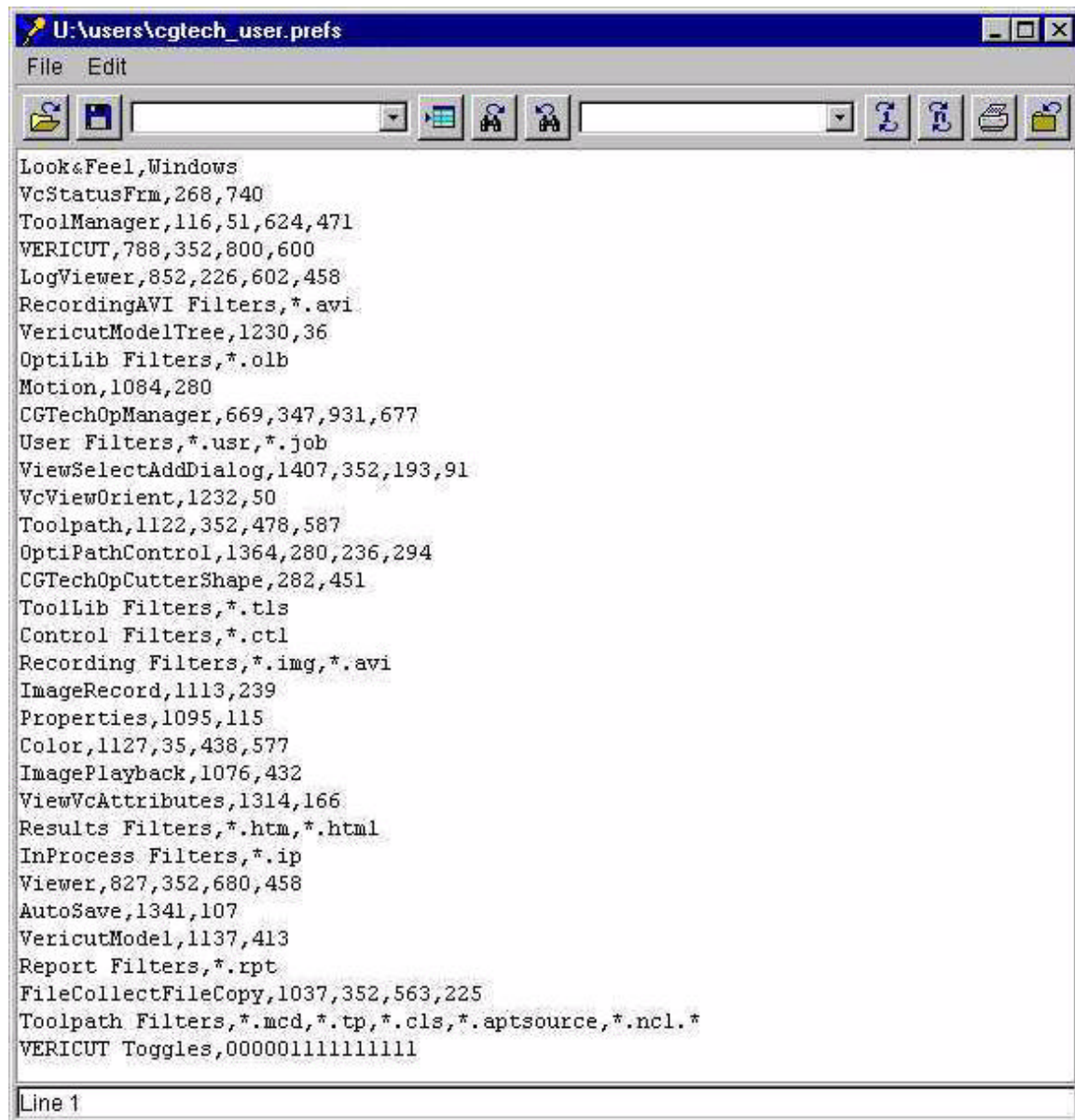
When users share a machine (or an account), CGTech recommends that each user set a unique CGTECH_USERNAME environment variable to prevent overwriting the Preferences file. See "Environment variables" in the *Reference* chapter for more information.

To reset Preferences file settings:

Delete the "cgttech_user.prefs" from your home directory.

On Windows systems, you can choose **Reset VERICUT Preferences file** from the CGTech programs group to accomplish this action.

Sample Preferences file "cgttech_user.prefs":



[Back to file types list](#)

Results file

Help on VERICUT

Typical file extension: .htm

An HTML file that contains overview information about the VERICUT session, such as: the current tool path and User files, part images (via **AutoSave**), tool images and machining times, optimization results, and links to the error Log file and Optimized tool path file (if any).

Exact contents of this vary, depending on what VERICUT was configured to do during the session. For example, if OptiPath was not turned on to optimize the tool path-then optimization results do not appear, if images were not automatically saved-then images do not appear, and so on.

Results files are created automatically and viewed via clicking **Info menu > Results File**. You can also view these files using your HTML browser.

Sample Results file:

VERICUT Report
Tuesday, July 25, 2000 10:28:57 AM PST

User File : C:\medopt1.mtr

Thumbnail	
	part.mtr seq 315
	medopt1 5.jpg ShadeCopy, Color change Ratio 2 number 9
	medopt1 10.jpg ShadeCopy, Color change Ratio 2 number 9
	medopt1 38.jpg ShadeCopy, Color change Ratio 2 number 9

Toolpath File : C:\medopt1.mcd

Seq #	Tool Change Record	Line #	Library ID	Height	Time
1	N001.T1.MG	9	1	1.5	2.39939
2	N023.T2.MG	33	2	2	4.20419
Total Time					6.60358

Number of Errors: 0
Number of Warnings: 0

Besides viewing the file directly, here are some other things you can do from the Results file:

View error Log file—Click on the "Log file" name link.

View full size images—Click on the thumbnail or image file name link. (Applicable only when **ShadeCopy** and **AutoSave** were configured to capture them.)

View optimized tool path file—Click on the OptiPath modified toolpath file name link. (Applicable only when **OptiPath** was configured to optimize.)

Help on VERICUT

[Back to file types list](#)

Status file

Typical file extension: .txt

An ASCII text file that contains whatever data is sent to the Status window during the simulation (ref. **Info menu > Status**). A Status file is created when the CGTECH_STATUS environment variable is set to a file name before VERICUT is executed. The file is overwritten when the VERICUT model is reset.

For each tool path record processed, the following status data is written:

REC—Indicates a tool path record processed.

REC fields—Toolpath Record Line Number, Toolpath Record

DATA—Specifies most of the status values separated by commas. Fields not displayed in the Status window are represented by commas.

DATA fields—Mch Axis Loc- X, Y, Z, A, B, C, U, V, W, Tool Tip Loc- X, Y, Z, I, J, K, Errors, Warnings, Time, Time%, Dist, - Dist%, Coolant, Feedrate, OP Feedrate, Spindle, OP Spindle

TOOLCHANGERE—The record which caused the last tool change (present only if the Status window is configured to show this field).

TOOLDESC—The current tool description (present only if the Status window is configured to show this field).

TOOLGEOMETRY—The current tool geometry (present only if the Status window is configured to show this field).

APT example:

```
REC,12,GOTO/3,2,1
```

```
DATA,,,,,,,,,,,,,3,,,,0,0,0.5355,,,,ON ,10 IPM,,500,CLW ,
```

```
TOOLCHANGERE,CUTTER/1.125,,25,0,0,0,0,2.1
```

G-code example:

```
REC,20,N0130G0X-.2Y-.3Z2.S1200M3
```

```
DATA,,,,,,,,,,,,,19,,,4,0,0,1.6979,,,,OFF,2 IPM,,1200,
```

```
TOOLCHANGERE,N0120T4M6
```

[Back to file types list](#)

Stereolithography model file

Typical file extension: .stl

A Stereolithography model file, also known as an “STL” or “SLA” file, is an ASCII text or binary file which describes virtually any surfaced or solid model shape. The STL data representing the model shape is composed of 3 sided facets with associated surface normals. STL files typically originate from a CAD system and represent an enclosed model used by an SLA machine, such as developed by 3D Systems.

Help on VERICUT

Stereolithography model files are loaded into VERICUT via the **Model menu > Model Definition** function. For an STL file to be used in VERICUT as a solid model, the STL data must represent a fully enclosed (watertight) model. If gaps exist between facets or facets overlap, display results are unpredictable. When STL files are used as surface models they do not have to be watertight.

When cutting is started or the model is analyzed, these symptoms indicate the STL file is not a watertight model:

- model is not displayed at all
- model is displayed with holes
- portions of the model are missing

Tips:

If any of the above symptoms are experienced, type using the **Stock Consistency Check** option (ref. **File menu > Properties: General** tab), or use PolyFix to improve the condition of the file.

Back to file types list

Stereolithography model file

Typical file extension: .stl

A Stereolithography model file, also known as an “STL” or “SLA” file, is an ASCII text or binary file which describes virtually any surfaced or solid model shape. The STL data representing the model shape is composed of 3 sided facets with associated surface normals. STL files typically originate from a CAD system and represent an enclosed model used by an SLA machine, such as developed by 3D Systems.

Stereolithography model files are loaded into VERICUT via the **Model menu > Model Definition** function. For an STL file to be used in VERICUT as a solid model, the STL data must represent a fully enclosed (watertight) model. If gaps exist between facets or facets overlap, display results are unpredictable. When STL files are used as surface models they do not have to be watertight.

When cutting is started or the model is analyzed, these symptoms indicate the STL file is not a watertight model:

- model is not displayed at all
- model is displayed with holes
- portions of the model are missing

Tips:

If any of the above symptoms are experienced, type using the **Stock Consistency Check** option (ref. **File menu > Properties: General** tab), or use PolyFix to improve the condition of the file.

Back to file types list

Tool Library file

Typical file extension: .tls

An ASCII text file that contains descriptions of cutting tools, shanks and holders, as well as other tool data used by VERICUT. Specific tool data is stored and retrieved via a tool "ID".

Help on VERICUT

This file is created and maintained via the **Setup menu > Tool Manager** function. The Tool Library file used by VERICUT when the **Setup menu > Toolpath** function is configured to do so.

Note: If errors occur when attempting to load the file, consult the *ERRORs and Problem Solving* chapter for assistance.

Back to file types list

Toolpath file

Typical file extension: .mcd (G-code), .tp (APT), many other extensions depending on CAM system

Toolpath, or "tool path" files are ASCII text files that contain data describing cutting tool positions, machine information, and other information required to operate NC machine tools.

Toolpath files to be simulated or optimized are referenced in VERICUT via the **Setup menu > Toolpath** function. VERICUT does not actually keep a copy of the tool path files, but instead stores tool path file names and keep tracks of which record in the file is being processed.

Tool paths files can be divided into two general types:

- APT-CLS tool path files

- G-code tool path files

A general discussion of each type follows.

APT-CLS tool path files

APT and CLS tool path files are generic file formats output by CAM systems that are intended to be easily read by the NC programmer. They are intermediate file formats that typically are not used directly by the NC machine. Before the data can be used by an NC machine these files must be post-processed to a G-code tool path file format containing codes specific to that machine tool (see "G-code tool path files" below).

VERICUT processes "Simple ASCII APT" tool paths-meaning all macros, motion commands, and motion copy routines are pre-processed into an ASCII centerline file containing GOTO, CIRCLE, CYCLE, and miscellaneous machine commands. No geometry definitions, symbolic substitutions, macros, or complex motions (such as GORGT, GOLFT, etc.) are present.

Common APT-CLS tool path variations that VERICUT can process are listed below.

- ACL

- CADRA CL data

- CATIA workstation APT

- CV APT

- ProManufacturing APT

- UG CLS

To process one of the APT-CLS tool path variations listed above, be sure to specify the appropriate **Toolpath Type** under **Setup menu > Toolpath**.

G-code tool path files

Help on VERICUT

G-code tool path files are formatted for use directly by the NC machine. These files typically originated as APT-CLS tool path files (see "APT-CLS tool path files" above), and have been post-processed to contain specific codes for a given machine tool.

To process G-code tool path file, be sure to specify **Toolpath Type=G-Code Data** under **Setup menu > Toolpath**. While VERICUT easily processes G-code data that conforms to the EIA Standard RS-274 format, other variations can also be simulated with a little configuration effort. **Setup menu > Control, Machine, and Modals** functions provide most of the configurability needed to support various NC machine and control combinations.

Back to file types list

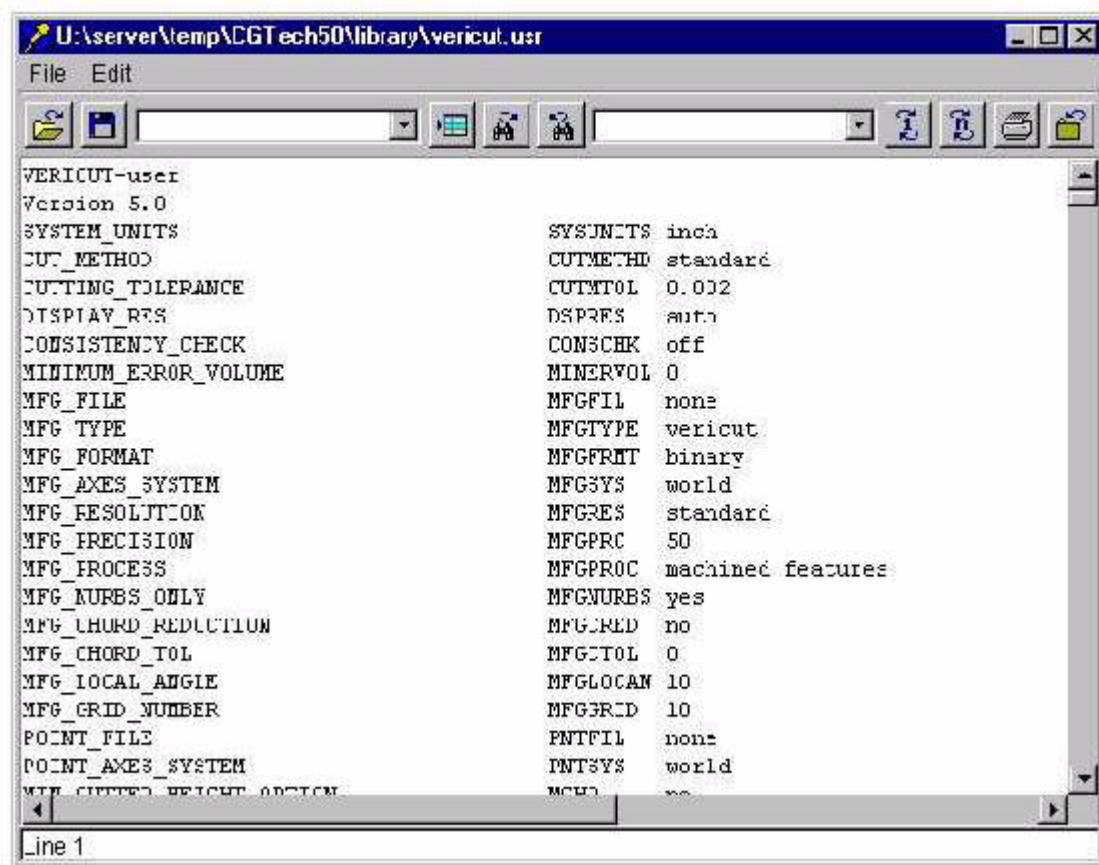
User file

Typical file extension: .usr

An ASCII text file that contains VERICUT session settings, also known as "user configuration values" or "user values", including: measurement units, colors, choices, data field values, directory paths and file names, etc.. These files are relatively small in size, yet can control the entire cutting process and minimize the effort required to configure the system for processing a tool path file.

User files are saved via the **File menu > Save As** function, and loaded via **File menu > Open**. User files can also be specified on the command line when VERICUT is executed, such as during batch processing to simulate tool paths without user interaction.

Sample User file "vericut.usr" (in CGTech library):



Help on VERICUT

Within the User file, the left column contains descriptive information about user variables, the center column lists the user variable names, and the right column lists saved values for corresponding user variables.

Use the user variable names listed in the center column when referencing via command line options. For more information, see "Command line options" in the *Reference* chapter.

Note: Environment variables can be used to affect VERICUT's use of User files. For more information, see "Environment variables" in the *Reference* chapter.

Back to file types list

VERICUT Linear-sweep file

Typical file extension: .swp

A VERICUT Linear-sweep file, or "Linear-sweep file" for short, is an ASCII text file that describes an extruded shape. A 2-D profile is defined in the XY plane, then swept (extruded) along the model's Z axis.

Linear-sweep files are loaded into VERICUT via the **Model menu > Model Definition** function, and contain the following data:

File type record—Specifies the type of VERICUT file, for example "VERICUT-model".

Version number record—Software version in which the file is to be used, for example "version-3.0".

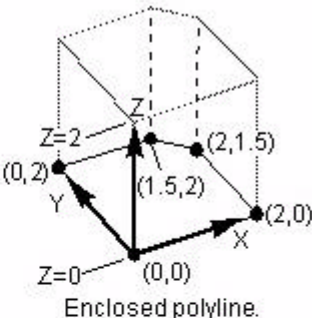
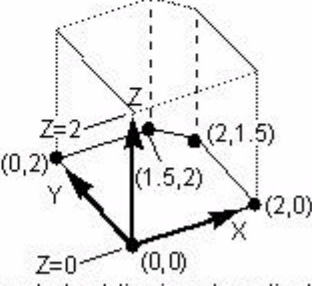
Model file identifier record—Model file type: "2" identifies a Linear-sweep file.

Sweep start value—Z value of the linear sweep start.

Sweep end value—Z value of the linear sweep end.

Polyline data—The remaining records in the file are the XY point pairs which define the polyline to extrude. Each X and Y value must be on a separate line with a space between values. There is no limit on the number of point pairs allowed, but there must be at least 3 pairs. The polyline can be completely enclosed or open ended, but can not cross over itself. Open ended polylines are automatically closed to the start point.

Sample Linear-sweep files:

Generic Linear-sweep File:	Sample Linear-sweep File 1:	Sample Linear-sweep Solid 1:
<pre> VERICUT-model version-3.0 2 Sweep start Z Sweep end Z x y x y . . x y </pre>	<pre> VERICUT-model version-3.0 2 0 2 0 0 2 0 2 1.5 1.5 2 0 2 0 0 </pre>	 Enclosed polyline.
Sample Linear-sweep File 2: <pre> VERICUT-model version-3.0 2 0 2 0 0 2 0 2 1.5 1.5 2 0 2 </pre>		Sample Linear-sweep Solid 2:  Open ended polyline is automatically closed to the start pt. The resulting solid is identical to solid 1.

See also the sample Linear-sweep file "vericut.swp" (in CGTech library).

[Back to file types list](#)

VERICUT Polygon file

Typical file extensions: design- .dsn, fixture- .fix, stock- .stk, any- .ply

A VERICUT Polygon file, or "Polygon file" for short, is an ASCII text file that describes virtually any shape open surface or enclosed shape. The polygon (faceted) data is composed of 3 or 4 sided polygons with associated surface normals that represent the true geometric model shape.

Polygon files are loaded into VERICUT via the **Model menu** > **Model Definition** function, and contain the following data:

File type record—Specifies the type of VERICUT file, for example "VERICUT-model".

Version number record—Software version in which the file is to be used, for example "version-3.0".

Model file identifier record—Model file type: "1" specifies a Polygon file.

Sweep start value—Z value of the linear sweep start.

Sweep end value—Z value of the linear sweep end.

Polygon format record—Describes the surface normal data and vertex data formats. This record contains three integer numbers separated by blanks: n1 n2 n3 (e.g. "0 0 1") where:

n1 (normal vector type)—Specifies the type of surface normal vector information present in the polygon file.

0- each polygon has one normal vector

1- each vertex has a normal vector

n2 (normal vector direction)—Specifies the surface normal vector direction relative to the model surface.

0- normal vectors point inward

1- normal vectors point outward

2- normal vector directions are inconsistent (should ONLY be used after both the “0” and “1” values have been attempted with unfavorable results; requires *significantly* more processing time and computer resources)

n3 (Polygon format type)—Specifies the polygon data format.

0- binary data format (recommended- requires less storage space and is processed faster, however, *binary formatted files are computer-specific*)

1- ASCII data format

Polygon data—The remaining records in the file are the polygon data records which define the model shape. Each model facet (or polygon) is represented by a series of records describing its number and location of vertices, as well as surface normal vector direction(s). There is no limit on the amount of data allowed.

Two polygon data formats are supported. The polygon data type must be consistent for the entire file.

Format 1- 1 normal per polygon:

vertex count (3 or 4)

X Y Z (vertex 1)

X Y Z (vertex 2)

X Y Z (vertex 3)

I J K (normal)

Format 2- 1 normal per vertex:

vertex count (3 or 4)

X Y Z (vertex 1)

X Y Z (vertex 2)

X Y Z (vertex 3)

I J K (normal 1)

I J K (normal 2)

I J K (normal 3)

Help on VERICUT

For a Polygon file to be used as a solid model in VERICUT, the polygon data must represent a fully enclosed "watertight" model. If gaps exist between facets or facets overlap, display results are unpredictable.

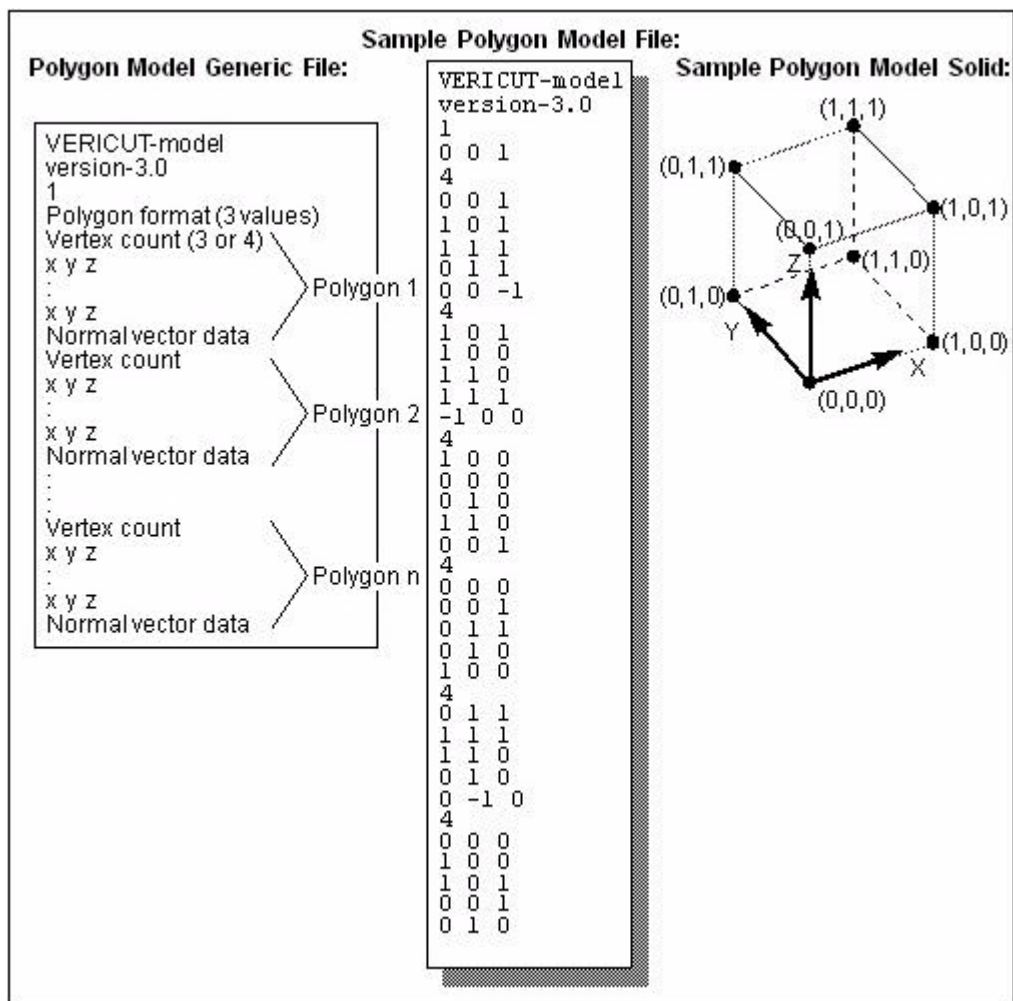
The symptoms listed below indicate that the Polygon model file does not represent a watertight model.

Model is not displayed at all

Model is displayed with holes

Portions of the model are missing

Sample Polygon file:



See also the sample Polygon file "vericut.dsn" (in CGTech library).

[Back to file types list](#)

VERICUT Solid-of-revolution file

Typical file extension: .sor

Help on VERICUT

A VERICUT Solid-of-revolution file, or "Solid-of-revolution file" for short, is an ASCII text file that describes a revolved shape. A 2-D profile is defined in the ZX plane, then revolved around the model's Z axis.

Solid-of-revolution files are loaded into VERICUT via the **Model menu > Model Definition** function, and contain the following data:

File type record—Specifies the type of VERICUT file, for example “VERICUT-model”.

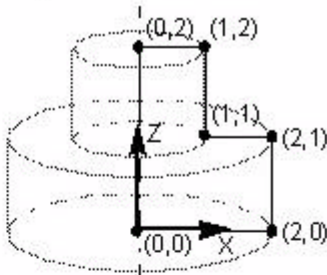
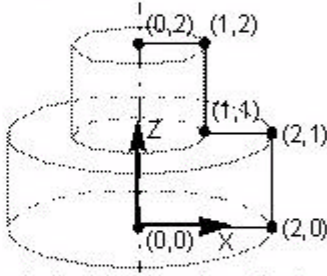
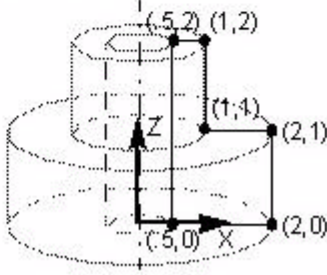
Version number record—Software version in which the file is to be used, for example “version-3.0”.

Model file identifier record—Model file type: “0” identifies a Solid-of-revolution file.

Tolerance of revolution—Chordal tolerance (intol) of the revolved surface, controls the S.O.R.’s degree of roundness for the model database.

Polyline data—The remaining records in the file are the XZ point pairs which define the polyline to revolve. Each X and Z value must be on a separate line with a space between values. There is no limit on the number of point pairs allowed, but there must be at least 3 pairs. X values must be greater than or equal to zero. The polyline can be completely enclosed or open ended, but can not cross over itself. Open ended polylines are automatically closed to the start point.

Sample Solid-of-revolution files:

S.O.R. Generic File: <pre> VERICUT-model version-3.0 0 tolerance of revolution x z x z . . . x z </pre>	Sample S.O.R. File 1: <pre> VERICUT-model version-3.0 0 .005 0 0 2 0 2 1 1 1 1 2 0 2 0 0 </pre>	Sample S.O.R. Solid 1:  Enclosed polyline includes the Z
	Sample S.O.R. File 2: <pre> VERICUT-model version-3.0 0 .005 0 0 2 0 2 1 1 1 1 2 0 2 </pre>	Sample S.O.R. Solid 2:  Open ended polyline is automatically closed along the Z axis to the start pt. The resulting solid is identical to solid 1.
	Sample S.O.R. File 3: <pre> VERICUT-model version-3.0 0 .005 .5 0 2 0 2 1 1 1 1 2 .5 2 .5 0 </pre>	Sample S.O.R. Solid 3:  Enclosed polyline does not include the Z-axis. The resulting solid is hollow.

See also the sample Solid-of-revolution file "vericut.sor" (in CGTech library).

[Back to file types list](#)

Supported APT-CLS tool path records

APT tool path records typically contain a major vocabulary word followed by a slash ("/"), then a combination of data and/or minor vocabulary words. APT records can be preceded by leading spaces or tabs. Spaces can appear anywhere in the record. APT record data typically ends prior to column 72. However, a "\$" at the end of a record continues onto the following line.

The APT tool path records processed by VERICUT are listed below in alphabetical order. See the appropriate section for details. *Records not listed are typically ignored by VERICUT.*

CATIA0 record

CIRCLE record

CLEARP record

CLRSRF record

CNTRL record

CONT record

COOLNT record

CUTTER record

CYCLE record

CYLNDR record

FEDRAT record

FROM record

GODLTA record

GOHOME record

GOTO record

HEAD record

KNOT record

LOAD TOOL record

LOADTL record

MODE record

MOVARC record

MSYS record

MULTAX record

Help on VERICUT

NURBS record
PPRINT record (general_purpose)
PPRINT/ATP CBOX record
PPRINT/TOOL_AXIS record
PPRINT/VERICUT record (or PPRINTVERICUT)
PPRINT/VERICUT-COMMAND record
PPRINT/VERICUT-CUTCOLOR record
PPRINT/VERICUT-MATRIX record
PPRINT/VERICUT-MODAL record
PPRINT/VERICUT-OPTIPATH record
PPRINT/VERICUT-TC record
RAPID record
RETRCT record
ROTABL record
ROTATE record
ROTHED record
SET_MODE record
SPINDL record
STOP record
SURFACE record
TLAXIS record
TLDATA record
TURRET record
UNITS record

CATIA0 record

Specifies a CATIA 12 parameter tool path transformation matrix.

This record is processed when **Process Matrix** is active. (Ref. **Setup menu** > **APT Settings: Motion** tab)

Example:

\$\$*CATIA0

Help on VERICUT

\$\$*AXS2				
\$\$	.00000	.00000	1.00000	-6.59040
\$\$	.99360	.11299	.00000	-16.50458
\$\$	-.11299	.99360	.00000	-3.32025

Back to APT/CLS record list

CIRCLE record

Causes circular motion. The GOTO point preceding the CIRCLE record defines the start of the circular motion. The GOTO following the CIRCLE defines the circular motion end. Helical motion is generated when the end point measured along the tool axis differs from the start point.

This record is processed when **Process Circles** is active. (Ref. **Setup menu > APT Settings: Motion** tab)

Formats:

CIRCLE/x,y,z,i,j,k,r[,n,n,n,n] - where "x,y,z" defines the circle center point, "i,j,k" is the axis orientation. "r" is the circle radius, but is not used since the radius is computed from the start, end and center points. The vector defines the positive Z-axis of counter-clockwise motion following the right-hand rule. In the 11 parameter format, VERICUT only uses the first 7 parameters.

Examples:

CIRCLE/1,2,3,0,0,1,5 - CCW circle motion about point 1,2,3, radius of 5

CIRCLE/1,2,3,0,0,-1,5 - as above, except CW arc motion

Back to APT/CLS record list

CLEARP record

Defines a clearance plane used by subsequent RETRACT records.

Formats:

CLEARP/OFF

CLEARP/value

CLEARP/XYPLAN(YZPLAN or ZXPLAN),value - establish a clearance plane parallel to the part coordinate XY, YZ, or ZX plane, respectively at "value" distance from the part coordinate system origin. Only one plane may be active at one time. If no plane option is specified, XYPLAN is assumed.

CLEARP/PERPTO,value - establishes a clearance plane perpendicular to the tool axis at "value" distance from the part coordinate system origin. Clearance plane distance is measured in the direction of the tool axis. Changing the tool axis vector changes the clearance plane.

Back to APT/CLS record list

CLRSRF record

Defines a clearance plane, same as CLEARP record.

Help on VERICUT

Back to APT/CLS record list

CNTRL record

Defines a control point for use with a NURBS record.

Formats:

CNTRL/x,y,z[,i,j,k][,WEIGHT,w] - where "x,y,z" are the control points (or poles) for tool position, optional "i,j,k" values control the tool axis, and optional "w" specifies the weights.

Back to APT/CLS record list

CONT record

GOTO continuation record, similar to GOTO record.

Formats:

CONT/x,y,z
x,y,z
:
CONT/x,y,z,i,j,k
x,y,z,i,j,k
:

Back to APT/CLS record list

COOLNT record

Controls coolant usage. Text following the slash is displayed in the Status window as the current coolant condition.

Formats:

COOLNT/anytext
COOLNT/ON[,anytext]
COOLNT/OFF

Back to APT/CLS record list

CUTTER record

Describes the shape of the cutting tool.

This record is processed by default, however, you can use "**Tool Change By**" to ignore these and process

Help on VERICUT

other records in the tool path file to retrieve cutting tool descriptions from a VERICUT tool library. (Ref. **Setup menu > Toolpath**)

Formats:

CUTTER/d,r,e,f,a,b,h

CUTTER/d,r,h

CUTTER/d,r (requires **Min. Cutter Height** > 0)

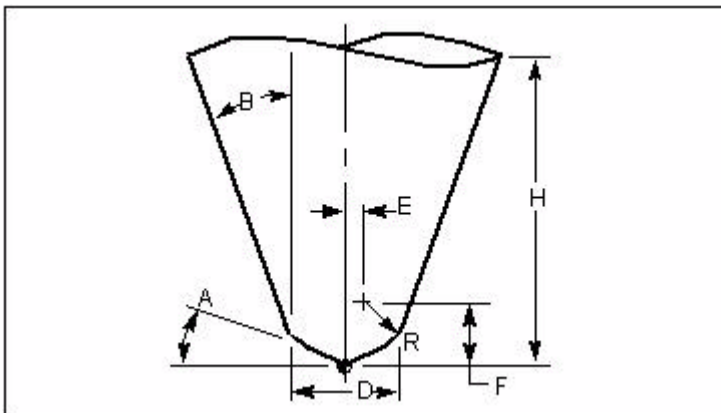
CUTTER/d (requires **Min. Cutter Height** > 0)

CUTTER/XCUT(SHANK or HOLDER),PT(PTS or ARC),profile descr.

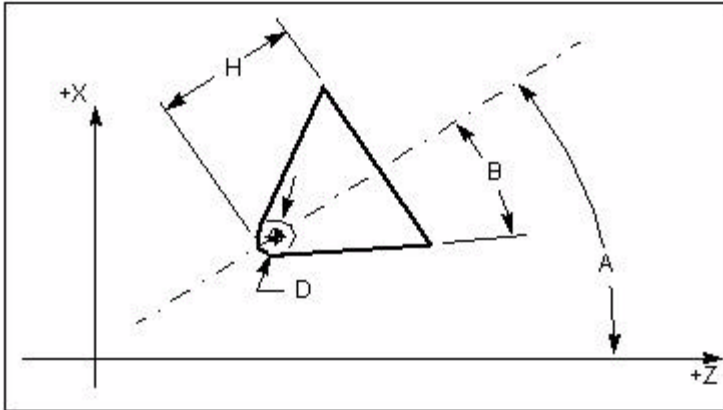
The table below and figures that follow describe how the values “d” through “h” are interpreted, based on if the cutter is used for milling or turning.

Value	Milling Cutter	Turning Cutter
d	diameter	nose diameter
r	corner radius	not used
e	dist. from tool centerline to corner radius center	not used
f	dist. from tool tip to corner radius center	not used
a	base angle	tool angle- degrees from spindle axis
b	side angle	side angle- degrees from tool centerline to its side
h	height	height- measured from tool tip along its centerline

Generic 7 Parameter Mill Cutter Definition: CUTTER/d,r,e,f,a,b,h



Generic 4 Parameter Turn Cutter Definition: CUTTER/d,0,0,0,a,b,h (or CUTTER/d,,,a,b,h)



See more APT-CLS records.

CYCLE record

Describes a canned tool axis cycle motion (e.g. drilling, reaming, boring, etc.) that is to be repeated at subsequent motion locations.

The **Drill Cycle** feature controls when and how these records are simulated. (Ref. **Setup menu > Motion**)

Formats:

CYCLE/<parameters> - where "<parameters>" provide information about how the canned motion is to be applied.

If canned cycle motions are not displayed as expected ensure that **Drill Cycle** is not set to **Ignore**.
If cycle simulation is still not correct, use **Cycle Def** to configure how VERICUT interprets canned cycle commands. (Ref. **Setup menu > APT_Settings:_Motion** tab)

Back to APT/CLS record list

CYLNDR record

Causes circular motion, same as CIRCLE record.

Back to APT/CLS record list

FEDRAT record

This record sets the feed rate, or speed at which the tool moves. Text following the slash is displayed in the Status window as the current feedrate condition.

By default, a zero feedrate is used until a FEDRAT record is processed. The active feed rate value is used until another FEDRAT record is encountered. A RAPID motion overrides the active FEDRAT for that motion only.

Formats:

FEDRAT/value[,IPM] also: IPR, MMPM, MMPR - where "value" represents feedrate in the current mode. The feedrate value, and mode if specified, are displayed in the Status window Feedrate field.

Help on VERICUT

Back to APT/CLS record list

FROM record

This record establishes where the tool is starting from. There is typically one record of this type per tool path file and it usually precedes any other motion record in the file. If a FROM record is not encountered, then the first GOTO point location is assumed as the from position.

Formats:

FROM/x,y,z[,i,j,k] - where "x,y,z" is the starting point and "i,j,k" represents the starting tool axis vector. If no ijk information is found, then a tool axis of 0,0,1 is assumed.

Back to APT/CLS record list

GODLTA record

This record designates incremental cutter movement relative to the current cutter location.

Formats:

GODLTA/x,y,z - specifies the incremental distance along each axis to move the tool from its current location. For example, if the tool is currently at 4,5,6 in xyz when GODLTA/1,2,3 is processed, the tool moves to x=5 (4+1), y=7 (5+2), and z=9 (6+3).

GODLTA/value - moves the cutter along the tool axis the specified amount.

If there is a transformation active, such as by a previous rotary motion command or a stock transformation, then the GODLTA values are transformed accordingly.

Back to APT/CLS record list

GOHOME record

This record specifies that the cutter return to the home position. The home position is assumed to be the location specified by the last FROM record processed. If a FROM record has not been processed, then 0,0,0 is used.

Formats:

GOHOME - move all axes to the home position. The **GOHOME Motion** feature controls if the cutter is moved in a straight line, or is retracted prior to going home. (Ref. **Setup menu > APT Settings: Motion** tab)

GOHOME/XAXIS(YAXIS or ZAXIS) - treated as GOTO statements with the specified AXIS element set to the FROM value for that axis. The other two elements are set to the current tool path position values.

Back to APT/CLS record list

GOTO record

Designates a position where the tool is moved to.

Formats:

Help on VERICUT

GOTO/x,y,z[,i,j,k] - where “x,y,z” represent the ending location of the tool motion and optional “i,j,k” values control the tool axis in multi-axis movements. “0,0,1” tool axis orientation is assumed. “GOTO” text is optional after the first GOTO motion, for example:

GOTO/x,y,z[,i,j,k]

x,y,z[,i,j,k]

:

Back to APT/CLS record list

HEAD record

Found in tool paths for mill-turn and 4-axis lathe machines, the “HEAD” record specifies which machining head is in use.

This record is processed for **Pro/MFG APT** tool path types. (Ref. **Setup menu > Motion**)

Formats:

HEAD/1- head 1 active

HEAD/2- head 2 active

HEAD/BOTH- head 1 active

HEAD/OFF- head 1 active

When head 2 is active the tool is changed via a “LOADTL” or “TURRET” record, VERICUT prepends the tool ID specified by the LOADTL/TURRET record with the characters “2_” to retrieve tools defined for use by head 2.

Example 1:

HEAD/1

TURRET/1

Looks for ID “1” in the Tool Library (interpreted as tool 1 for use by head 1).

Example 2:

HEAD/2

TURRET/1

Looks for ID “2_1” in the Tool Library (interpreted as tool 1 for use by head 2).

Back to APT/CLS record list

KNOT record

Defines a knot parameter for use with a NURBS record.

Format:

KNOT/t1,...,t(n-p+2) - where “t1” - “t(n-p+2)” are the knot parameters.

Help on VERICUT

Back to APT/CLS record list

LOAD/TOOL record

Specifies the tool number to be changed into the spindle, same as LOADTL record.

This record is not processed by default, however, you can use "**Tool Change By**" to process this record and retrieve cutting tool descriptions from a VERICUT tool library. (Ref. **Setup menu > Toolpath**)

Format:

LOAD/TOOL,n[,anytext]

See also HEAD record for information about how the Pro/MFG APT "HEAD" record affects how this record is used to retrieve tool from a VERICUT tool library.

Back to APT/CLS record list

LOADTL record

Specifies the tool number, or pocket number of the tool to be changed into the spindle. Optionally, this record may also specify tool length or other aspects about the tool, however, VERICUT ignores all parameters following the tool number.

This record is not processed by default, however, you can use "**Tool Change By**" to process this record and retrieve cutting tool descriptions from a VERICUT tool library. (Ref. **Setup menu > Toolpath**)

Format:

LOADTL/n[,anytext] - where "n" is the index number of the tool (number or alpha-numeric). Leading zeros in the tool ID are also ignored, for example "LOADTL/01" is interpreted as "LOADTL/1". Any text following the "/" or number data after tool ID is ignored, for example "LOADTL/2,LENGTH,6.500" is interpreted as "LOADTL/2".

See also HEAD record for information about how the Pro/MFG APT "HEAD" record affects how this record is used to retrieve tools from a VERICUT tool library.

Back to APT/CLS record list

MODE record

This record specifies the machining to be performed: mill or turn.

This record is processed for **Pro/MFG APT** tool path types. (Ref. **Setup menu > Motion**)

Formats:

MODE/MILL(TURN)

Back to APT/CLS record list

MOVARC record

Causes circular motion, same as CIRCLE record.

Help on VERICUT

Back to APT/CLS record list

MSYS record

Specifies a 9 parameter tool path transformation matrix.

This record is processed when **Process Matrix** is active. (Ref. **Setup menu** > **APT Settings: Motion** tab)

Back to APT/CLS record list

MULTAX record

Determines if subsequent GOTO motions are multi-axis motions. The multi-axis condition is modal. When multi-axis mode is on it remains on until another MULTAX record turns it off.

Formats:

MULTAX/ON

MULTAX/OFF

Back to APT/CLS record list

NURBS record

Defines a NURBS B-spline motion command. NURBS motion commands also utilize the CNTRL record and KNOT record as part of the B-spline definition.

Format:

NURBS/[p]- where “p” indicates the B-spline order. The “NURBS” record typically spans multiple lines and includes a number of other record types, such as “KNOT”, “CNTRL”, etc.

Example:

```
NURBS/  
KNOT/1.0000000  
CNTRL/61.2424, 93.2577, -31.6260  
:  
CNTRL/59.8775, 93.2224, -31.6263
```

Back to APT/CLS record list

PPRINT record (general purpose)

Comment record that typically has no affect on tool motion. “PPRINT” resides in columns 1-6. Text following “PPRINT” is typically ignored by VERICUT, however, some PPRINT records can perform VERICUT actions. See other “PPRINT” records following this section for more information.

Back to APT/CLS record list

PPRINT/ATP CBOX record

Defines a stock block via min/max corner points.

Help on VERICUT

This record is processed by VERICUT when a stock model is loaded and the model is not yet cut. A new stock block is defined and fit into all displayed views.

Format:

PPRINT/ATP CBOX xmin,ymin,zmin,xmax,ymax,zmax

Example:

PPRINT/ATP CBOX -1,-2,-1,5,7,2 -defines a 6 x 9 x 3 block positioned X= -1, Y= -2, Z= -1

Back to APT/CLS record list

PPRINT TOOL AXIS record

Specifies a new tool axis orientation in a CATIA tool path. The tool orientation remains in effect until another record changes the tool axis orientation.

This record is processed for **CATIA APT** tool path types. (Ref. **Setup menu > Motion**)

Example:

PPRINT TOOL AXIS= XAX=-1.0000 YAX=.0000 ZAX=.0000

Back to APT/CLS record list

PPRINT/VERICUT record

(or PPRINTVERICUT) Sets a VERICUT user value, or performs a VERICUT command or action. PPRINT/VERICUT text must be between columns 1-72.

PPRINT data can span multiple PPRINT records via ending each PPRINT to be continued in a dollar sign. The next line must be a PPRINT/VERICUT record with the same keyword as the previous line. Words in PPRINT data can not be split by the dollar sign. Examples are provided in the sections that follow.

See one of these sections for information about specific PPRINT/VERICUT record formats:

PPRINT/VERICUT-COMMAND record

PPRINT/VERICUT-CUTCOLOR record

PPRINT/VERICUT-MATRIX record

PPRINT/VERICUT-MODAL record

PPRINT/VERICUT-MOTION record

PPRINT/VERICUT-OPTIPATH record

PPRINT/VERICUT-TC record

Back to APT/CLS record list

PPRINT/VERICUT-COMMAND record

Performs VERICUT commands or operations.

Help on VERICUT

Formats:

PPRINT/VERICUT-COMMAND auto-diff - performs an AUTO-DIFF comparison, then restores the cut model.

PPRINT/VERICUT-COMMAND pause n - adds “n” seconds of delay to an Image file being recorded. (See Image Record function)

PPRINT/VERICUT-COMMAND repaint - repaints solid objects with colors assigned via the **File menu > Colors: Assign** tab.

PPRINT/VERICUT-COMMAND reset_cutcolor - resets all of the model colors to the "**Reset Cut Color**" color. (Ref. **File menu > Colors: Assign** tab)

PPRINT/VERICUT-COMMAND reverse - reverses the VERICUT model.

PPRINT/VERICUT-COMMAND display - updates FastMill display with current cuts.

PPRINT/VERICUT-COMMAND display_fixtures(remove_fixtures) - displays (or removes) all fixture models.

PPRINT/VERICUT-COMMAND load_model_by_name <modelname> - loads a model by its name.

PPRINT/VERICUT-COMMAND remove_model_by_name <modelname> - removes a model by its name.

PPRINT/VERICUT-COMMAND load_fixture_by_seq <n> - loads the fixture referenced in the sequential order of defined fixtures. For example, “PPRINT/VERICUT-COMMAND load_fixture_by_seq 2” loads the second defined fixture.

PPRINT/VERICUT-COMMAND remove_fixture_by_seq <n> - removes the fixture referenced in the sequential order of defined fixtures.

The following examples load the fixture model “Model 2 (Fixture)” from the sample models list:

PPRINT/VERICUT-COMMAND load_model_by_name “Model 2”

PPRINT/VERICUT-COMMAND load_fixture_by_seq 1

Model Definition List

Model 1 (Stock)

Model 2 (Fixture)

Model 3 (Design Solid)

Back to APT/CLS record list

PPRINT/VERICUT-CUTCOLOR record

Assigns the color referenced by index “n” to be the cut color for the current tool. The index number of any defined color can be specified (Ref. **File menu > Colors: Define** tab). Color index numbers range from 0-n where “0” refers to the first color in the list of defined colors, “1” refers to the second, and so on. The “PPRINT/VERICUT-CUTCOLOR” index number is always 1 less than the index seen in the user interface.

Help on VERICUT

For example, "PPRINT/VERICUT-CUTCOLOR 2" assigns "Blue" as the cut color for the current tool from the color list below:

Color Definition List

Red

Green

Blue

Cyan

Back to APT/CLS record list

PPRINT/VERICUT-MATRIX record

Specifies a 12 parameter tool path transformation matrix.

This record is processed when **Process Matrix** is active. (Ref. **Setup menu** > **APT Settings: Motion** tab)

Format:

PPRINT/VERICUT-MATRIX I1,J1,K1,D1,I2,J2,K2,D2,I3,J3,K3,D3

Examples:

PPRINT/VERICUT-MATRIX 1,0,0,2,0,1,0,3,0,0,1,-1

Translates the tool path 2 units in X, 3 in Y, and -1 in Z.

PPRINT/VERICUT-MATRIX 1,0,0,2,\$

PPRINT/VERICUT-MATRIX 0,1,0,3,\$

PPRINT/VERICUT-MATRIX 0,0,1,-1

As above, except uses the "\$" continuation format.

PPRINT/VERICUT-MATRIX -1,0,0,0,0,1,0,0,0,0,-1,0

Rotates the tool path 180 degrees about the Y axis.

Back to APT/CLS record list

PPRINT/VERICUT-MODAL record

Specifies a user variable value: data field entry or option choice. One user variable/value can be set per record. Most user variables found in a User file can be set or modified in this way.

Formats:

PPRINT/VERICUT-MODAL <uservariablename>=<uservalue>

Example setting the **Fast Feed** value to "100":

Help on VERICUT

PPRINT/VERICUT-MODAL FASTFEED=100

Example controlling the **Machining Type** for proper material removal simulation:

PPRINT/VERICUT-MODAL MACHTYPE=milling

<milling machine motions>

PPRINT/VERICUT-MODAL MACHTYPE=turning

<turning machine motions>

Back to APT/CLS record list

PPRINT/VERICUT-MOTION record

Controls when motions and cutting are simulated.

Formats:

PPRINT/VERICUT-MOTION off(on)- turns motion simulation and cutting off (or on).

When off, motion commands and tool positions are processed, but simulation, cutting, collision detection, and axis overtravel detection are turned off during these moves. When motion is turned on, the machine/tool “jumps” to its current position. The motion “on” condition is automatically set when a reset is done.

Back to APT/CLS record list

PPRINT/VERICUT-OPTIPATH record

Controls tool path optimization. Use this record to specify which optimization record to use, suspend optimization, or turn optimization back on again.

See also OptiPath Control or OptiPath Manager functions in the *Reference* chapter.

Formats:

PPRINT/VERICUT-OPTIPATH off

Suspends tool path optimization. When off, VERICUT continues to write unoptimized tool path records to the optimized tool path file. Optimization must be resumed via “PPRINT/VERICUT-OPTIPATH on” before the next tool change.

PPRINT/VERICUT-OPTIPATH on

Resumes optimization turned off via “PPRINT/VERICUT-OPTIPATH off”.

PPRINT/VERICUT-OPTIPATH DESC="my_optipath_tool_description" TEETH=n

where "my_optipath_tool_description" is an optimization record description which could contain any characters, including delimiters. The description must be enclosed in double quotes. "N" is the number of cutting teeth.

This record specifies two of the four values needed to identify which optimization record to use. The other two values come from **Material** and **Machine** on the **OptiPath menu > Control** window. If an optimization record matching the four values is not found, VERICUT continues

Help on VERICUT

to write unoptimized tool path records to the optimized tool path file.

Back to APT/CLS record list

PPRINT/VERICUT-TC record

Defines a cutter, shank or holder shape. Interpreted like a CUTTER record, anything allowed to follow "CUTTER/" is also allowed to follow "PPRINT/VERICUT-TC".

Like "CUTTER/", this record is processed by default, however, you can use "**Tool Change By**" to ignore these and process other records in the tool path file to retrieve cutting tool descriptions from a VERICUT tool library. (Ref. **Setup menu > Toolpath**)

Examples:

```
PPRINT/VERICUT-TC 1.0,.125,0,0,0,0,3
```

Defines an APT 7 parameter cutter.

```
PPRINT/VERICUT-TC XCUT,PTS,0,0,.5,0,ARC,1,0,.5,PTS,1,.5,1,1
```

Defines a special cutter profile.

```
PPRINT/VERICUT-TC XCUT,PTS,0,0,.5,0,$
```

```
PPRINT/VERICUT-TC ARC,1,0,.5,PTS,1,.5,$
```

```
PPRINT/VERICUT-TC 1,1
```

As above, except uses the "\$" continuation format.

```
PPRINT/VERICUT-TC SHANK,PTS,0,0,.5,0,.5,1,0,1
```

Defines a shank profile.

```
PPRINT/VERICUT-TC HOLDER,PTS,0,0,1,0,1.5,.5,1.5,3,0,3
```

Defines a holder profile.

Incorrect examples:

```
PPRINT/VERICUT-TC XCUT,PTS,0,0,.5,0,ARC,$
```

```
PPRINT/1,0,.5,PTS,1,.5,1,1,0,1
```

Continuation record missing "VERICUT-TC".

```
PPRINT/VERICUT-TC XCUT,PTS,0,0,.5,0,AR$
```

```
PPRINT/VERICUT-TC C,1,0,.5,PTS,1,.5,1,1,0,1
```

The word "ARC" is split by "\$".

```
PPRINT/VERICUT-TC XCUT,PTS,0,0,.5,0,1,.5,1,1 $
```

```
PPRINT/VERICUT-COMMAND reverse
```

Continuation record does not have the same keyword "TC".

Help on VERICUT

Back to APT/CLS record list

PPRINT equivalents in G-code format

VERICUT can process PPRINT comments that have been post-processed into G-code format, for example “(VERICUT-...)”. All comment text must be between columns 1 and 72. The formats and capabilities of this record are similar to those described for the PPRINT/VERICUT record, however, the G-code version substitutes “PPRINT/” with a “Begin Comment” character and ends with a “Close Comment” character. Examples follow.

Example comment records that reverse the model display:

APT tool path: **PPRINT/VERICUT-COMMAND reverse**

Fanuc control: **(VERICUT-COMMAND reverse)**

Cincinnati Acramatic control: **(MSG,VERICUT-COMMAND reverse)**

RAPID record

Overrides the current feed rate and causes a machine tool to move as fast as possible. When the RAPID record is processed by VERICUT, the motion following the RAPID moves the tool using the **Fast Feed** feed rate. (Ref. **Setup menu** > **Motion** tab)

For example, assume the file below is processed while **Fast Feed=100**.

RAPID

GOTO/1,2,3

FEDRAT/20

GOTO/2,3,4

RAPID

GOTO/3,4,5

GOTO/4,5,6

The motions are simulated by VERICUT as follows:

feed rate from the current tool location to 1,2,3: 100

feed rate from 1,2,3 to 2,3,4: 20

feed rate from 2,3,4 to 3,4,5: 100

feed rate from 3,4,5 to 4,5,6: 20

In both examples below, motion from 1,2,3 to 4,5,6 uses a feed rate of “20”:

Example 1:

FROM/1,2,3

RAPID

Help on VERICUT

FEDRAT/20

GOTO/4,5,6

In example 1, a FEDRAT record immediately follows the RAPID record before a motion is encountered. Under this condition, the FEDRAT value overrides a RAPID record.

Example 2:

RAPID

FROM/1,2,3

FEDRAT/20

GOTO/4,5,6

In example 2, a FROM record is not considered to be a motion record. Therefore, the RAPID record has no affect.

Back to APT/CLS record list

RETRCT record

Retracts the cutter along the tool axis to a specified clearance plane, or by a specified amount.

Formats:

RETRCT- retract the tool along the tool axis until it reaches the clearance plane defined by the previous CLEARP or CLRSRF record

RETRCT/value- retract along the tool axis the specified amount

RETRCT/ON- retract to the Z level prior to a cycle record.

Example:

GOTO/1,2,3

RETRCT/ON

CYCLE/DRILL,1.0,3,IPM,.1 \$\$ Retracts to Z=3 between cycle positions.

Back to APT/CLS record list

ROTABL record

Specifies an amount of table rotation. Unless otherwise specified, rotation about the Y-axis is assumed.

This record is processed when **ROTABL** is active. (Ref. **Setup menu > APT Settings: Rotary** tab)

Formats:

ROTABL/AAXIS(BAXIS or CAXIS),value

ROTABL/AAXIS(BAXIS or CAXIS),ATANGL,value

ROTABL/AAXIS(BAXIS or CAXIS),INCR,value

Help on VERICUT

ROTABL/ATANGL,value

ROTABL/INCR,value

Back to APT/CLS record list

ROTATE record

Specifies an amount of table rotation, same as ROTABL record.

This record is processed when **ROTATE** is active. (Ref. **Setup menu > APT Settings: Rotary** tab)

Formats:

ROTATE/AAXIS(BAXIS or CAXIS),value

ROTATE/AAXIS(BAXIS or CAXIS),ATANGL,value

ROTATE/AAXIS(BAXIS or CAXIS),INCR,value

ROTATE/ATANGL,value

ROTATE/INCR,value

Back to APT/CLS record list

ROTHED record

Specifies an amount of head rotation.

This record is processed when **ROTHED** is active. (Ref. **Setup menu > APT Settings: Rotary** tab)

Formats:

ROTHED/ATANGL,value

ROTHED/value

Back to APT/CLS record list

SET/MODE record

This record specifies the machining to be performed: mill or turn.

This record is processed for **UG CLS** tool path types. (Ref. **Setup menu > Motion**)

Formats:

SET/MODE,MILL

SET/MODE,TURN

Back to APT/CLS record list

SPINDL record

Help on VERICUT

Specifies spindle rotation speed (RPM); direction and gear ranges may also be specified. Text following the slash is displayed in the Status window as the current spindle condition.

Formats:

SPINDL

SPINDL/rpm[,anytext]

SPINDL/[anytext,]rpm

Back to APT/CLS record list

STOP record

Stops the NC machine at its current location. This record commonly indicates an operator function follows, such as: changing the tool, changing the part-fixtured setup, etc.

Back to APT/CLS record list

SURFACE record

Causes circular motion, same as CIRCLE record.

Back to APT/CLS record list

TLAXIS record

Specifies a new tool axis orientation. The new orientation is modal and is applied to following GOTO points that do not have I,J,K values.

This record is processed for **CATIA APT** tool path types. (Ref. **Setup menu > Motion**)

Example:

TLAXIS/ .000000, 1.000000, .000000

Back to APT/CLS record list

TLDATA record

This record describes the cutter shape and is processed with similar considerations as described for CUTTER record.

Formats:

TLDATA/MILL <parameters> - all mill record formats are supported.

TLDATA/DRILL <parameters> - all drill formats are supported, except drills having 180 degree tip angles.

TLDATA/TURN <parameters> - most turning TLDATA record formats are supported.

NOT supported:

TLDATA/GROOVE <parameters>

Help on VERICUT

TLDATA/THREAD <parameters>

[Back to APT/CLS record list](#)

TURRET record

Similar to LOADTL record, this record often appears in tool paths for turning operations to specify the tool number to be changed into location for cutting.

This record is not processed by default, however, you can use "**Tool Change By**" to process this record and retrieve cutting tool descriptions from a VERICUT tool library. (Ref. **Setup menu > Toolpath**)

The number following "TURRET" is often a composite of two numbers: a tool number and offset number. However, VERICUT simply interprets the entire number as a tool number. For example, while "TURRET/0101" may indicate tool 01 is used with offset 01), VERICUT interprets this to mean tool 0101 (or 101) is to be used from the VERICUT tool library.

Formats:

TURRET/n[,anytext]

TURRET/FACE,n[,anytext]

See also HEAD record for information about how the Pro/MFG APT "HEAD" record affects how this record is used to retrieve tools from a VERICUT tool library.

[Back to APT/CLS record list](#)

UNITS record

Sets the unit measurement system of the data in the tool path file: inches or millimeters.

Formats:

UNITS/INCHES

UNITS/MM

[Back to APT/CLS record list](#)

Control files

This section describes Control files available in the CGTech library ("library" directory in your VERICUT installation). These files enable you to quickly configure VERICUT to simulate how features of various NC controls affect the machining process. A Control file is loaded into VERICUT via the **Setup menu > Control > Open** function, or by opening a User file configured to use the control.

NC controls are listed below in alphabetical order, by control manufacturer and model. See also the accompanying notes file for details (same base file name with .txt extension-located in the "library" directory). You can do this in on-line Help by clicking on file names that appear as hyperlinks (underlined text).

Note: When installed, customized machines and controls can be found in the "samples" directory in your VERICUT installation. See "Sample-demo files" in this chapter for more information.

Control manufacturer:	Control model:	Control file name:
Acramatic (by Cincinatti)	850MC	acr850mc.ctl
	950MC	acr950mc.ctl
	CNC PC	acrcncpc.ctl
Allen-Bradley	8200	ab_8200.ctl
Bosch	CC300	bshcc300.ctl
Charmilles	200	chr200.ctl
Cincinatti - see "Acramatic"		
Fadal	CNC 88, Format 1 ("Fadal" mode)	fad88.ctl
	CNC 88, Format 2 ("Fanuc" mode)	fad88a.ctl
Fanuc	0M	fan0m.ctl
	0T	fan0t.ctl
	5M	fan5m.ctl
	6M	fan6m.ctl
	6T	fan6t.ctl
	9M	fan9m.ctl
	9T	fan9t.ctl
	10M	fan10m.ctl
	10T	fan10t.ctl
	11M	fan11m.ctl

Help on VERICUT

	11T	fan11t.ctl
	12M	fan12m.ctl
	12T	fan12t.ctl
	15M	fan15m.ctl
	15T	fan15t.ctl
	16i	fan16im.ctl
	16M	fan16m.ctl
	16T	fan16t.ctl
	16W	fan16w.ctl
	18i	fan18im.ctl
	18M	fan18m.ctl
	18T	fan18t.ctl
	21i	fan21im.ctl
	150M	fan150m.ctl
	150T	fan150t.ctl
	160i	fan160im.ctl
	160M	fan160m.ctl
	160T	fan160t.ctl
	180i	fan180im.ctl
	180M	fan180m.ctl
	180T	fan180t.ctl
	210i	fan210im.ctl
	3000C	fan3000c.ctl
Fidia	M30	fidm30.ctl
GEMINI - see "Kearney & Trecker"		
General Electric (GE)	2000MC CNC	ge_2000m.ctl
Generic controls (Fanuc-like)	decimal	generic.ctl (in), genericm.ctl (mm)
	leading zeros	genlead.ctl (in), genleadm.ctl (mm)

Help on VERICUT

	trailing zeros	gentrail.ctl (in), gentralm.ctl (mm)
Giddings & Lewis (G&L) - see "NumeriPath"		
Haas	CNC	hascnc.ctl
Heidenhain	Mill Plus	heimplus.ctl
	TNC 355 (plain language)	heid355.ctl
	TNC 407 (conversational)	hei407.ctl
	TNC 407 (ISO)	hei407g.ctl
	TNC 415b (conversational)	hei415b.ctl
	TNC 415b (ISO)	hei415bg.ctl
	TNC 415c (conversational)	hei415c.ctl
	TNC 415c (ISO)	hei415cg.ctl
	TNC 425 (conversational)	hei425.ctl
	TNC 425 (ISO)	hei425g.ctl
	TNC 426 (conversational)	hei426.ctl
	TNC 426 (ISO)	hei426g.ctl
Kearney & Trecker (K&T)	C	kt_c.ctl
	GEMINI-D	gemini_d.ctl
	GEMINI-E	gemini_e.ctl
Maho - see "Phillips"		
Makino	Professional Series 3	makpro3.ctl
Mazak	Mazatrol M-32	mazm32.ctl
	Mazatrol T-32	mazt32.ctl
NUM	1060M	num1060m.ctl
NumeriPath (by G&L)	800M	nph800m.ctl
	8000L	nph8000l.ctl
	8000M	nph8000m.ctl
Okuma	OSP700L	osp700l.ctl
	OSP5000L	osp5000l.ctl

Help on VERICUT

	OSP5000M	osp5000m.ctl
	OSP7000L	osp7000l.ctl
	OSP7000M	osp7000m.ctl
Olivetti	MC8600	oli8600.ctl
Phillips	CNC 432	phi432.ctl
	CNC 532	phi532.ctl
Sinumerik (by Siemens)	3T	sin3t.ctl
	810D	sin810d.ctl
	840C	sin840c.ctl
	840D	sin840d.ctl
	880M	sin880m.ctl
	FM-NC	sinfmnc.ctl
Sundstrand	MicroSWINC LM	swilm.ctl
	MicroSWINC M200	swim200.ctl
Tosnuc (by Toshiba)	888	tos888.ctl
Yasnac	MX-3	yasmx3.ctl

Machine files

This section describes Machine files available in the CGTech library ("library" directory in your VERICUT installation). These files enable you to quickly configure VERICUT to simulate how various NC machines will perform. A Machine file is loaded into VERICUT via the **Setup menu > Machine > Open** function, or by opening a User file configured to use the machine.

NC machines are listed below in alphabetical order, by manufacturer and model. See also the accompanying notes file for details (same base file name with .txt extension-located in the "library" directory). You can do this in on-line Help by clicking on file names that appear as hyperlinks (underlined text).

Note: When installed, customized machines and controls can be found in the "samples" directory in your VERICUT installation. See "Sample-demo files" in this chapter for more information.

Machine manufacturer:	Machine model:	Machine file name:
Aerostar	H2200 5ax horiz. mill	aerh2200.mch
Bohle	BW120S 5ax horiz. mill	bohbw120.mch
Bostomatic	405 5ax vert. mill	bos405.mch
Charmilles	Robofil 200 4ax wire EDM	chr200.mch
Cincinnati	T30 5ax horiz. mill	cint30.mch (in), cint30m.mch (mm), cint30_2.mch, cint30_3.mch
Deckel Maho - see "Maho"		
Dixi	100 4ax horiz. mill	dix100m4.mch
	100 5ax horiz. mill	dix100m5.mch
	150 4ax horiz. mill	dix150m4.mch
	150 5ax horiz. mill	dix150m5.mch
	350 TCA50S 5ax horiz. mill	dix350m.mch
Droop and Rein	DR969 mill	dr_dr969.mch
Fadal	VMC15 3ax vert. mill	fadvmc15.mch
	VMC4020 4ax vert. mill	fad4020.mch
	VMC 6030 3ax vert. mill	fad6030.mch
Fidia	218 5ax vert. mill	fid218.mch (in), fid218m.mch (mm)
Forest	Flexiax 510 5ax horiz. mill	forfx510.mch
Generic 3-D machines	2ax horiz. lathe	2hl.mch
	2ax horiz. turret lathe	2hl02.mch, 2hl03.mch

	4ax horiz. dual turret lathe	4hl02.mch, 4hl03.mch
	3ax vert. mill	3vm.mch (in), 3vmm.mch (mm), g3vm.mch, g3vmgl.mch (in), g3vmglm.mch (mm), g3vmtlc.mch (in), g3vmtlcm.mch (mm), g3vmtt.mch (in), g3vmttm.mch (mm),
	4ax horiz. mill	4hmtb.mch, g4hmtb.mch (table B)
	4ax horiz. mill/turn	4ht.mch (in), 4htm.mch (mm)
	4ax vert. mill	g4vmta.mch (table A)
	5ax horiz. mill	g5hmhatb.mch (head A, table B), g5hmtatb.mch (table A on table B),
	5ax vert. mill	g5vmhahb.mch (head A on head B), g5vmtahb.mch (table A, head B)
	Fanuc-controlled 3ax mill	fanuc3x.mch (in), fanuc3xm.mch (mm)
	Fanuc-controlled 4ax mill	fanuc4x.mch (in), fanuc4xm.mch (mm)
Giddings & Lewis (G&L)	G60 4ax horiz. boring mill	gl_g60.mch
HAAS	MiniMill 3ax vert. mill	hasmini.mch
	VB-1 5ax vert. mill	hasvb1.mch
Ingersoll	Mastercenter 4ax gantry mill	ingmast1.mch (detailed), ingmast2.mch (simple)
Kearney & Trecker (K&T)	HB4 4ax horiz. mill	kt_hb4.mch
	Model 180 4ax horiz. mill	kt_180.mch
Kuraki	KBM11 5ax horiz. boring mill	kurkbm11.mch
Maho	DMU 80 P 5ax horiz. mill	mahdmu80.mch
	MH700 3ax vert. mill	mah700m.mch (in), mahmh700m.mch (mm)
Makino	A55 Plus 4ax horiz. mill	maka55.mch
	A77 4ax horiz. mill	maka77.mch
	MC1010 5ax horiz. mill	mak1010.mch
Marwin	Advanced manufacturing cell 5ax horiz. mill	mrwamc.mch
Matsuura	FX-5 4ax vert. mill	matfx5.mch
	MAM 600HF vert. mill	mat600hf.mch

Help on VERICUT

Mazak	H500 4ax horiz. mill	mazh500.mch
	Super 400H 4ax horiz. mill	maz400h.mch
Mori Seiki	Partner M-400C1 5ax vert. mill	ms_m400.mch
Okuma	LB25 2ax horiz. lathe	okulb25.mch
SNK	FSP-100V 5ax horiz. mill	snkfsp01.mch
	SH-200 5ax gantry mill	snksh200.mch
Sundstrand	Omnimil OM3 5ax vert. mill	sunom3.mch
	Omnimil OM80 5ax horiz. mill	sunom80.mch
Toyoda	FH80 5ax horiz. mill	tydfh80.mch
Tsugami	MA3 4ax horiz.mill	tsuma3.mch
Warner & Swasey	Titan 4ax horiz. dual turret lathe	ws_titan.mch

Help on VERICUT

Sample-demo files

This topic is still being written.

Chapter 5 Training Sessions

Session 1- Introduction to the VERICUT user interface

This session shows how to define multiple views, orient views and how to navigate in the VERICUT file structure via the shortcuts. Using the **Shortcut** feature on file selection windows provides quick access to files in your Working Directory or CGTech library and sample files. Choosing a shortcut automatically filters for files in that directory.

For this session and the subsequent training sessions it is highly suggested to create a training directory to avoid overwriting any sample files installed with VERICUT.

Session Steps:

1. Execute VERICUT.

Access VERICUT on a Windows computer—From the Taskbar, click **Start Programs** **CGTech VERICUT 5.x VERICUT** where “5.x” is the desired VERICUT version number.

Access VERICUT on a UNIX computer—Type “vericut”. If the command is not found, include the full path, e.g. “/ <cgtech5x/<computertype/commands/vericut” where <cgtech5x represents the directory path to where CGTech products were installed, and <computertype is the type of computer on which VERICUT is being operated, for example: hp, rs, sgi, or solaris.

Note: VERICUT automatically loads the “vericut.usr” user file from the CGTech library

2. Define a Working Directory.

File > Working Directory

Make sure that the path is pointing to a directory where you have write permission e.g. your home directory.

Change directory if required.

OK

You can also quickly access or save files in this directory via the **Working Directory** entry in the file selection window Shortcut list.

3. Open a user file.

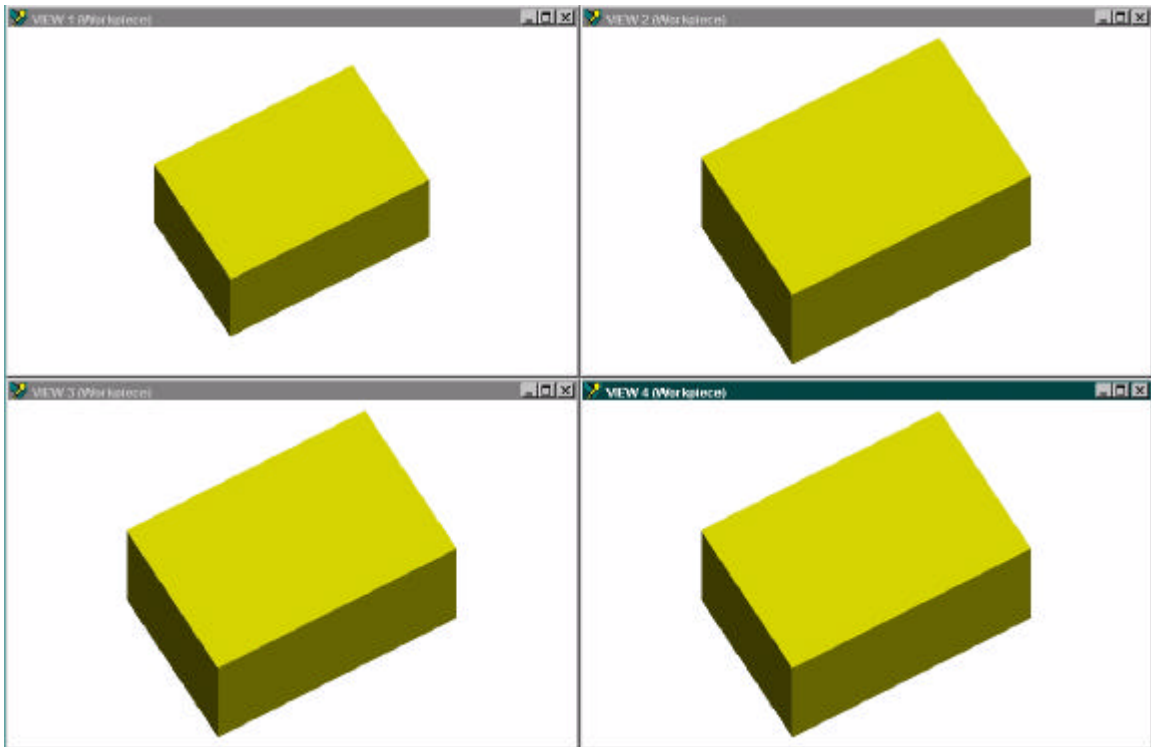
File > Open: Shortcut = CGTECH_SAMPLES

Click on **cardhold.usr**

Open (If a message appears, press **Reset.**)

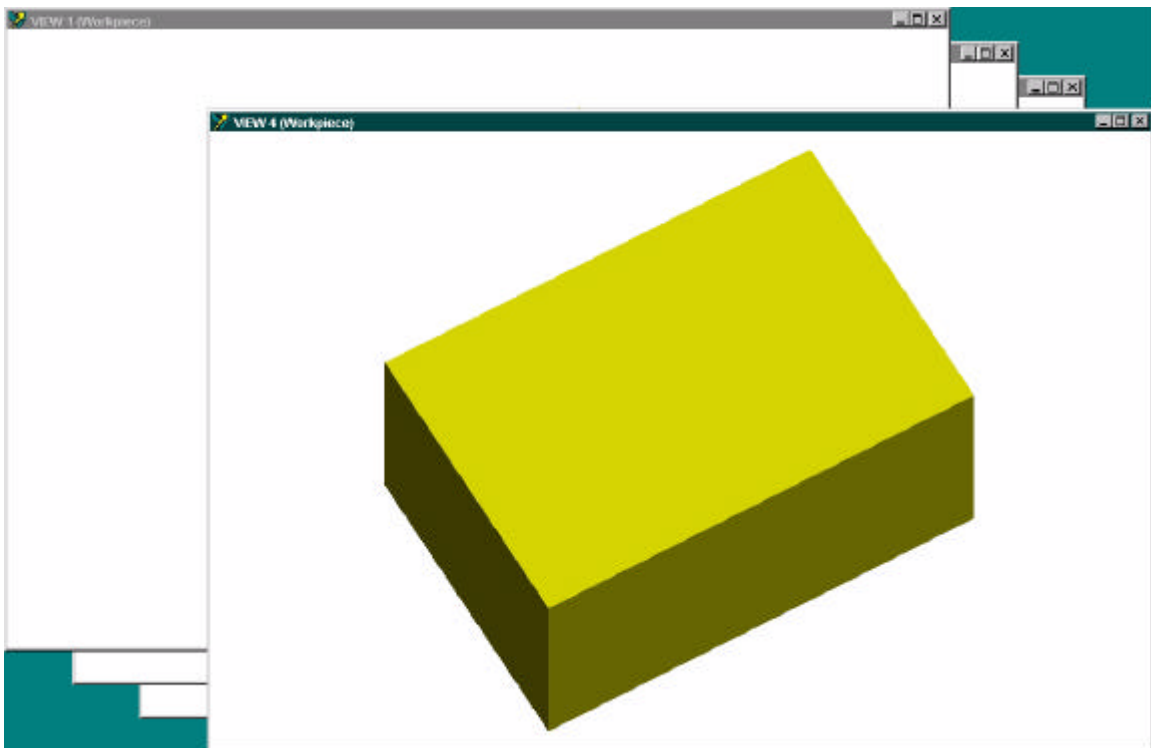
4. Change view layout.

View > Layout > Standard > Click on 4 views



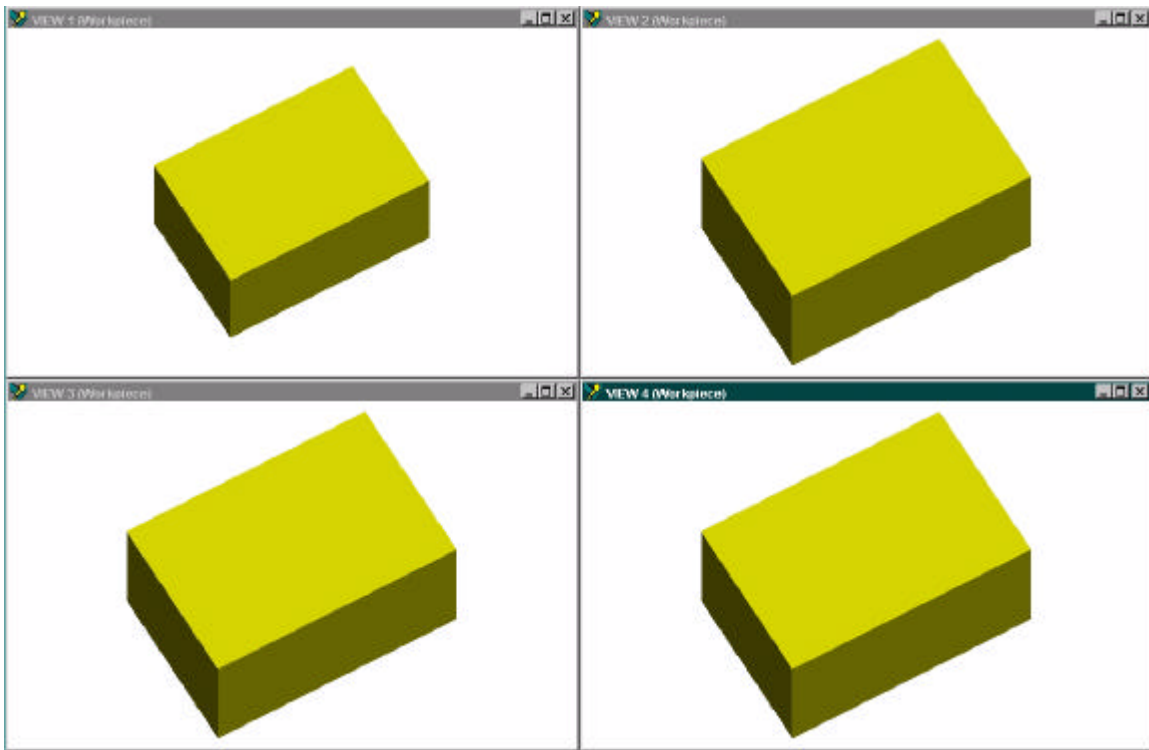
5. Change views to be cascaded (overlapped).

View > Layout > Cascade



6. Change views layout back to a tiled layout.

View > Layout > Layout > Tile Horizontally.



The top of the active view is highlighted, in all other views the top is gray.

7. Change the orientation of the views.

Left click in **VIEW 2** to make it the active view

View > Orient, XY

Increment= 30

Y-, X-, Fit

Left-click in **VIEW 3**

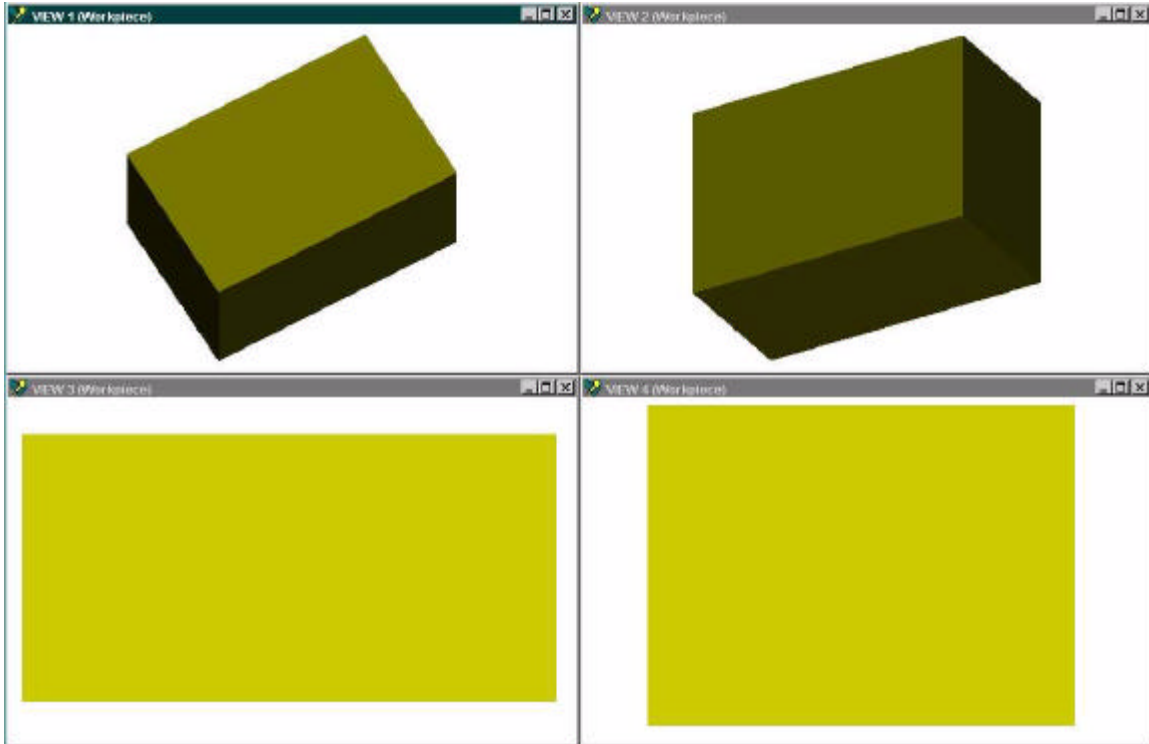
XZ

Left-click in **VIEW 4**

ZY

Increment= 90, Z+

Fit All



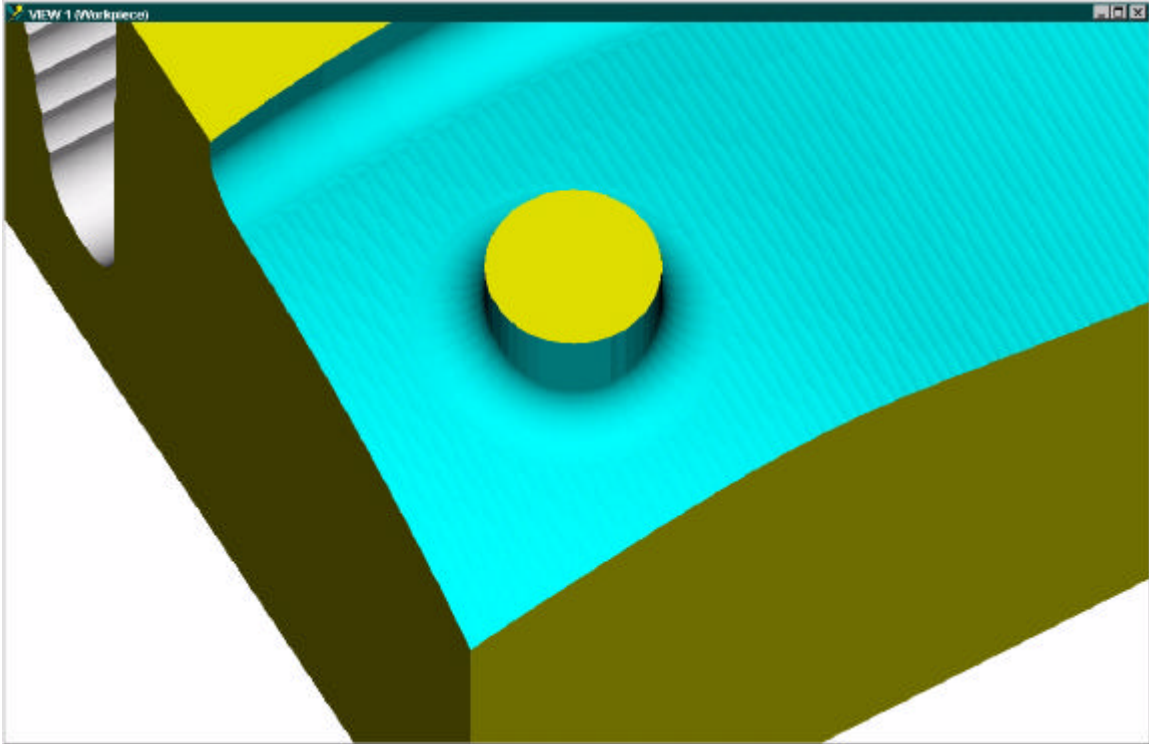
8. Cut the workpiece

Play to end 

9. Re-orient the cut workpiece

Left-click in **VIEW 1**

Zoom - Left-click top left of the boss, drag the box down. Then left-click to activate the zoom.



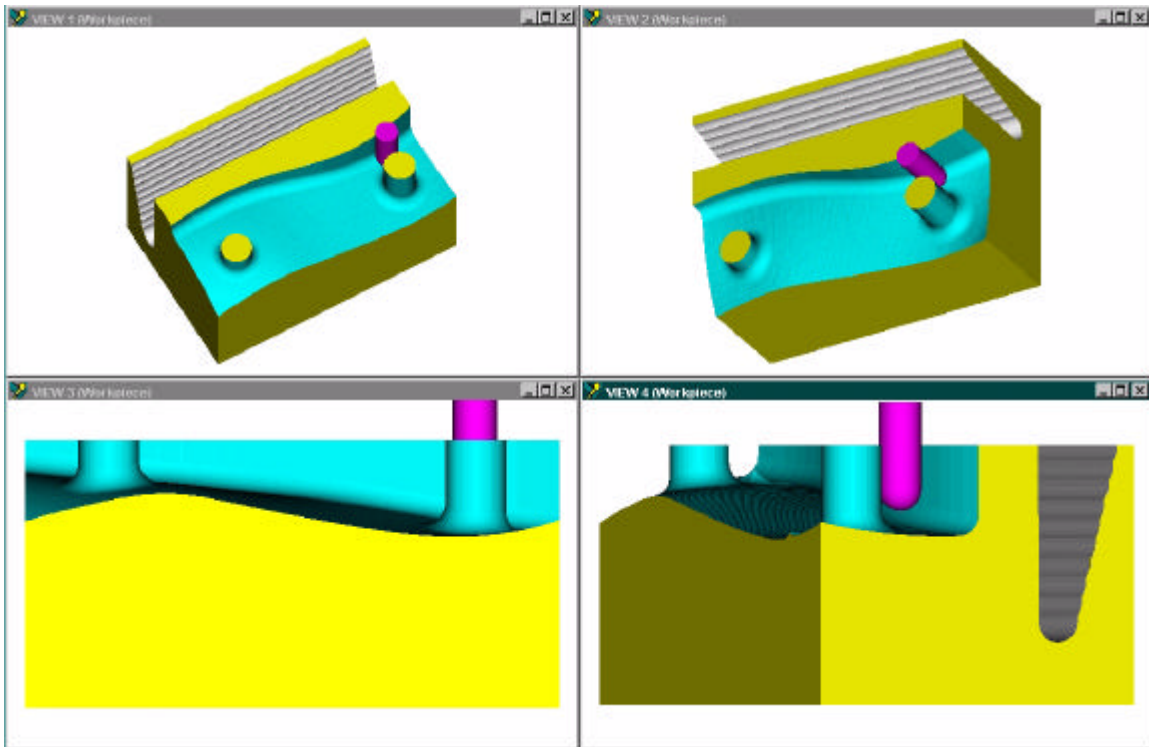
Left-click in **VIEW 4**

Reverse 

Refine Display 

10. Dynamically orient the part

Dynamic Y Rotation - In the active view (View 4), drag the cursor to the right to rotate the part as shown below. 



Click on **Dynamic Y Rotation** to de-activate it.

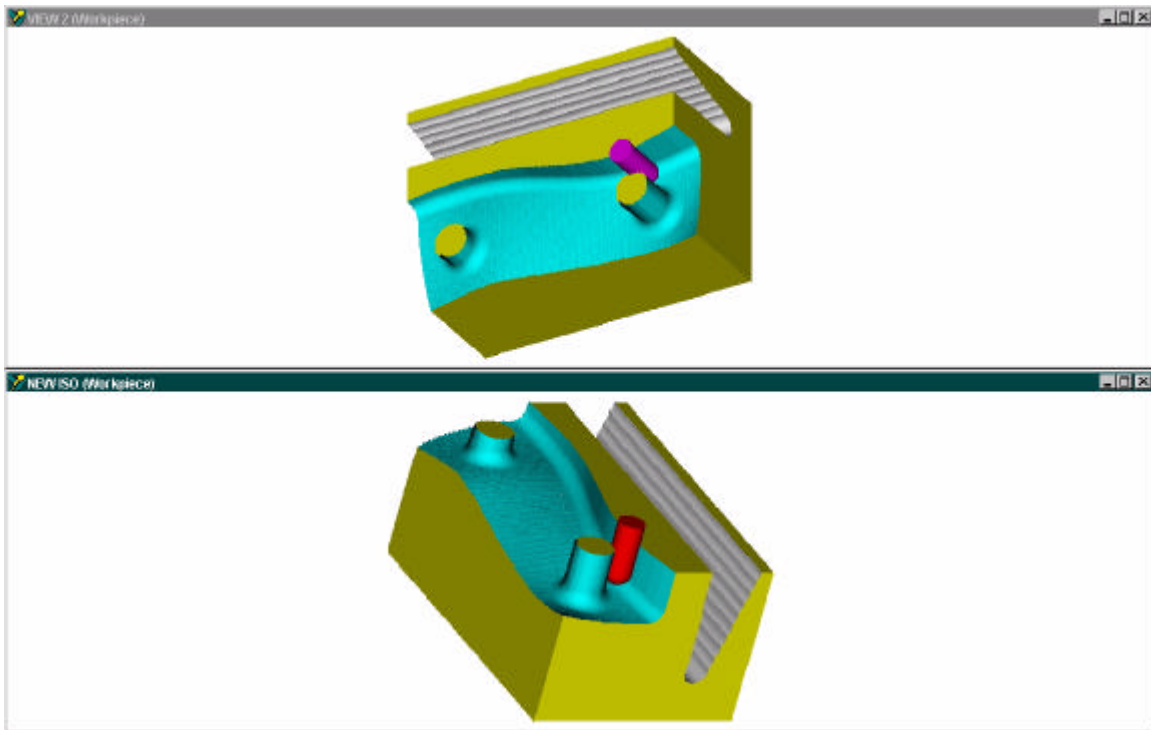
11. Close View 1,3 & 4

Close views using the "X" icon on the top right of each view.



12. Add a new view

View > Layout > Add View



13. Orient the new view (VIEW 1)

Left click in the lower view (VIEW 1)

View > Orient, YZ

Increment=30

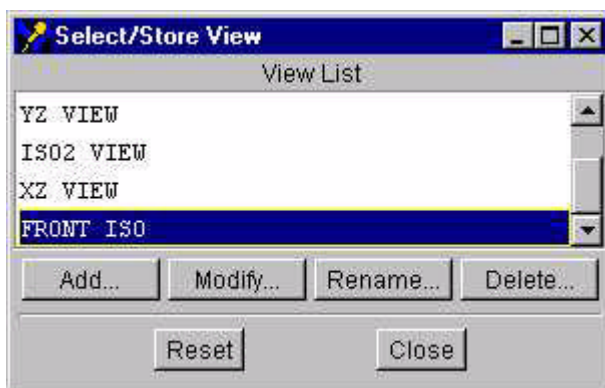
X+, Y+, Fit

Close

14. Store the new view orientation

View > Select/Store

Add: View Name= FRONT ISO, OK



Left-click in the upper view (VIEW 2)

Help on VERICUT

In the “**Select/Store View**” window, click on **XZ VIEW**

Click on **FRONT ISO**

Left-click in **VIEW 1**

Click on **ISO2 VIEW**

Close

A User file is an ASCII text file containing the VERICUT session configurations. Saving a User file enables you to quickly re-configure VERICUT for additional tool path processing, as well as displaying models and model images.

15. Save a new User File named “session1.usr”

File > Save As

Shortcut=Working Directory

File Name=Session1.usr, Save

16. Exit the VERICUT session

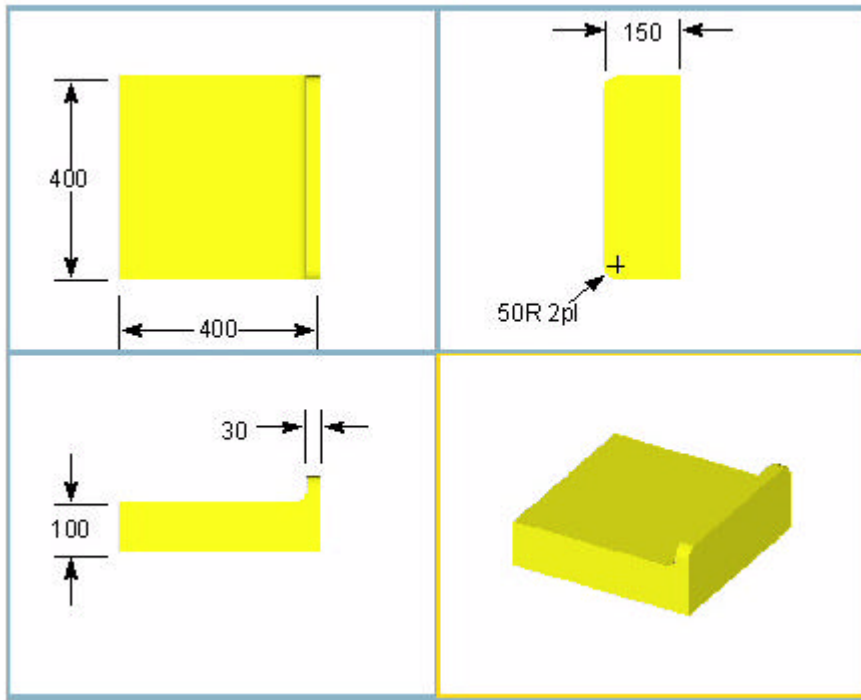
File > Exit

This session demonstrated how to define a working directory, load a User File, and how to orient and create views. The step prior to exiting showed how to save the VERICUT session configuration in a user file.

Session 2- Modeling a casting & fixture

This session shows how to use modeling features to assemble a stock model and a fixture with clamps. Knowledge of modeling is useful when CAD models of shapes that are not available, or the resources are not available to define them in a CAD system and/or transfer them to VERICUT. The 2D drawing below shows the standard drafting views of a metric casting shape to be defined.

Sample casting dimensions:



Session Steps:

1. Start with a new millimeter User file.

File > Properties

Default units=Millimeter, OK

File > New (A window may open prompting you to save the changes to the existing User file if required)

2. Add a Stock model to be a 400 x 400 x 100 mm block.

Model > Model Definition: Model tab

Type=Block, Length(X)=400, Width(Y)=400, Height(Z)=100, Add

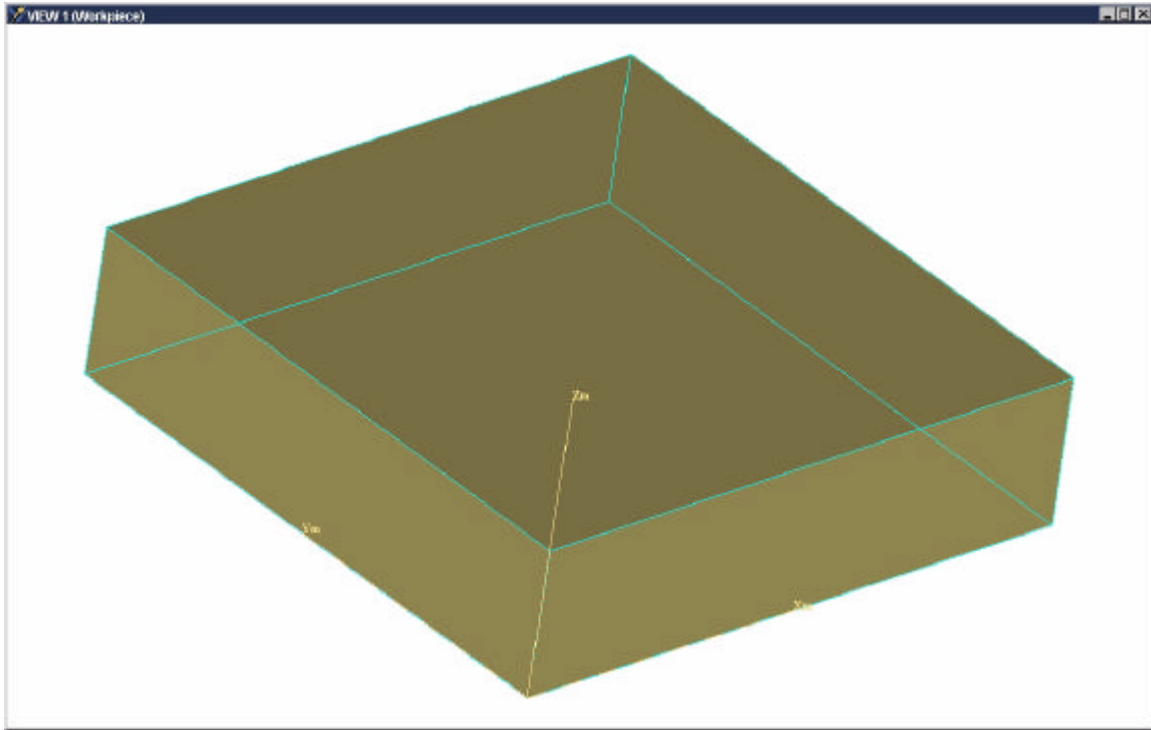
Fit 

3. Display the model axis system.

View > Axes

Select **Model**

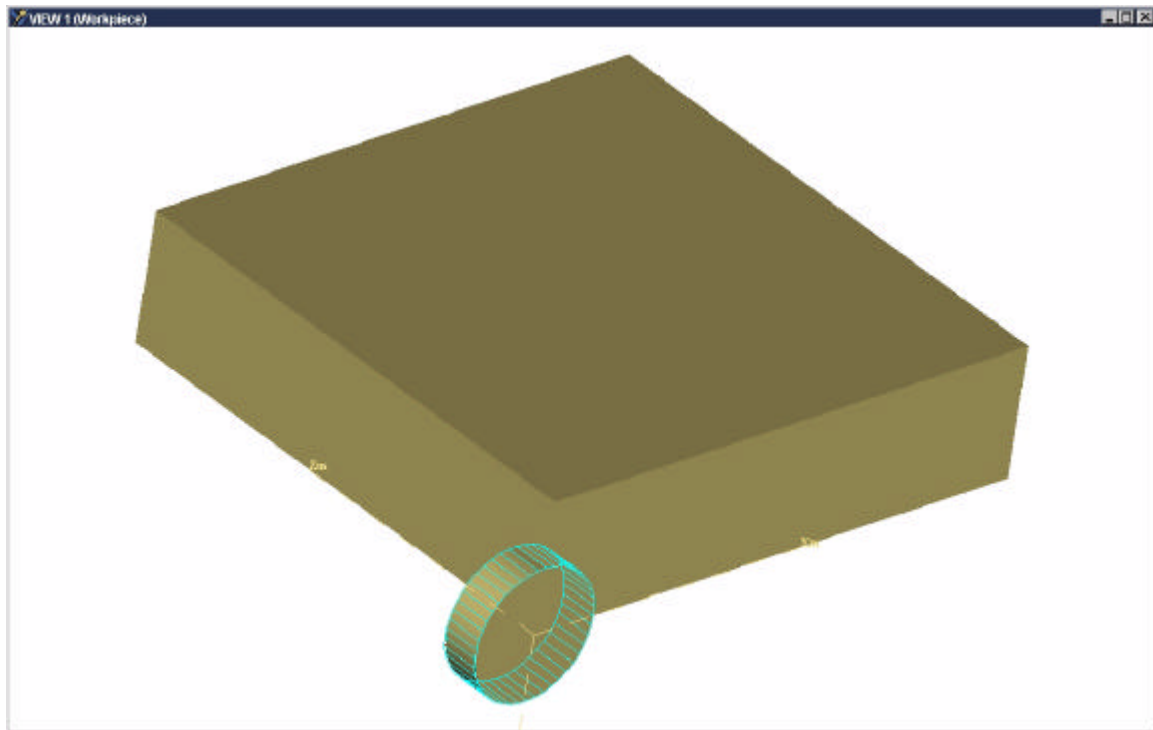
Close



4. Add cylinders to represent the two 50R corners of the standing web.

Type=Cylinder, Height=30, Radius=50, Add

Fit 



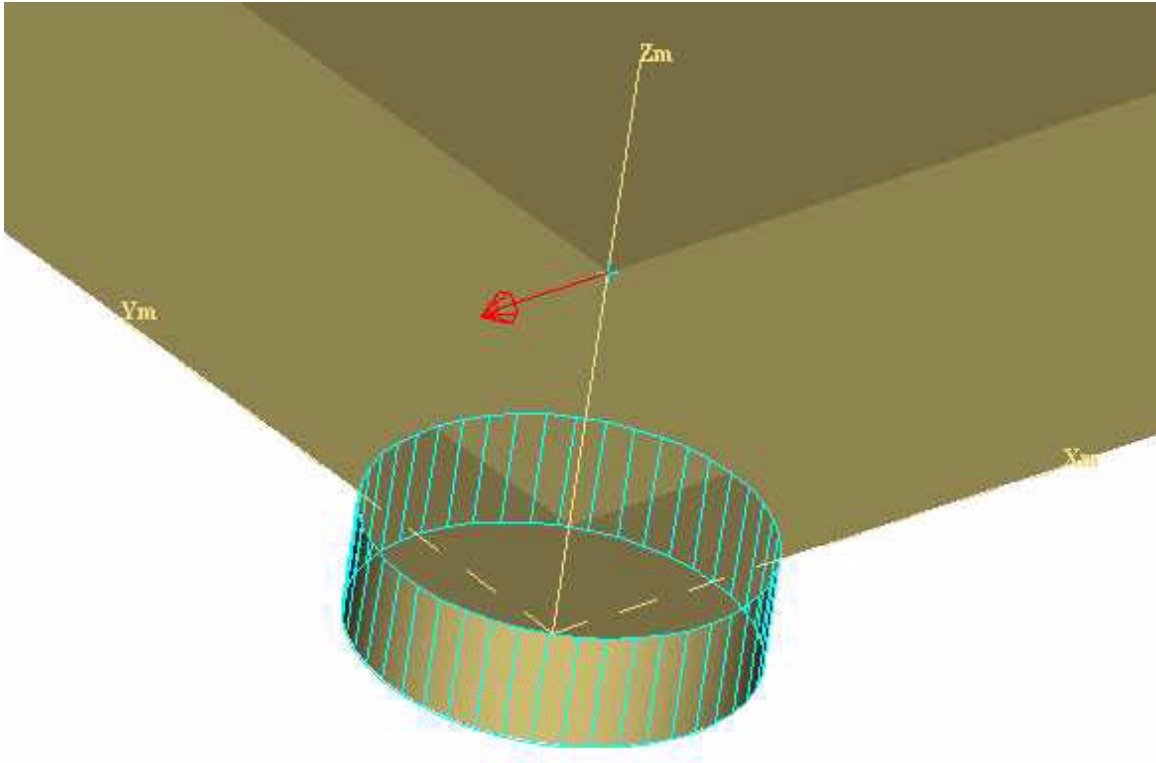
Click **Position** tab

Rotate Increment=90, X-

Tip: If the applied translation is incorrect, press **Undo** to return the object to its previous location.

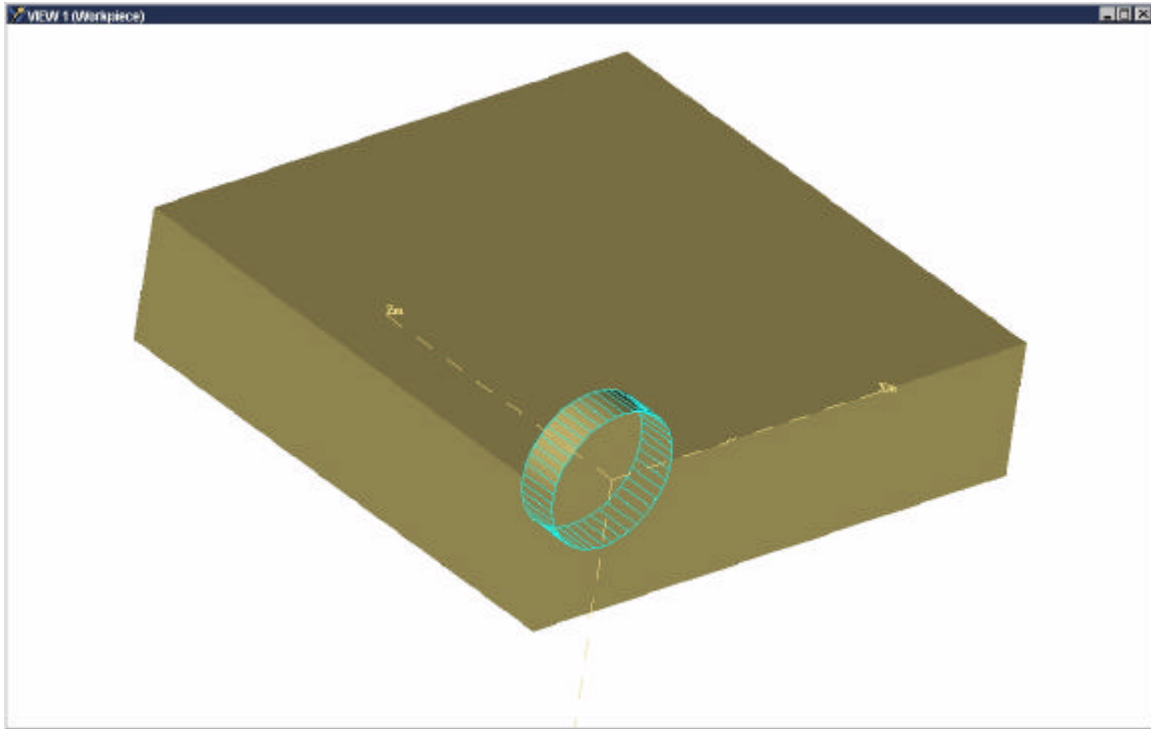
Translate From=0 0 0

Click **To:** Select top of the part, bottom left corner (value should be 0 0 100)

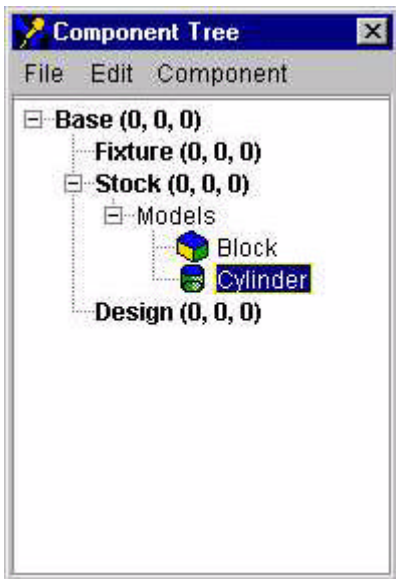


Move

Translate From=0 0 0, To=50 (or To=50 0 0), Move



Model Component Tree (Cylinder should be highlighted)

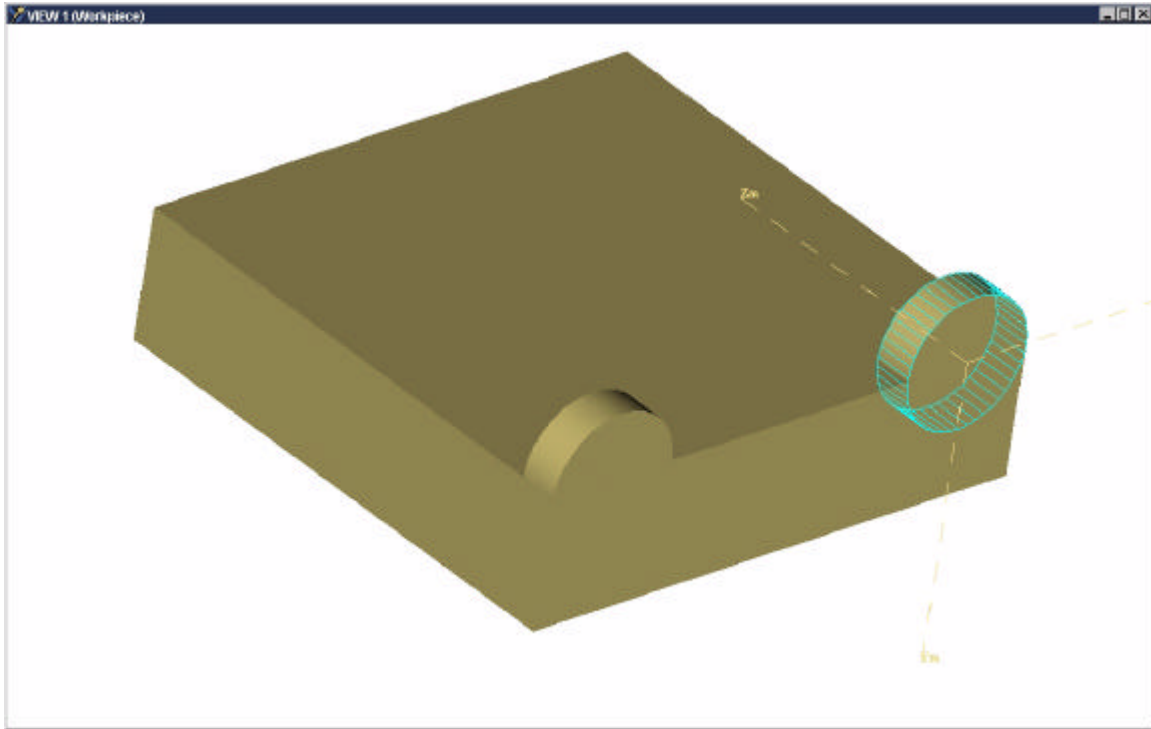


Right-click on the **Cylinder** model, **Copy**

Right-click on the **Cylinder** model, **Paste**

Click **Position** tab

Position=350 0 100, **Apply**



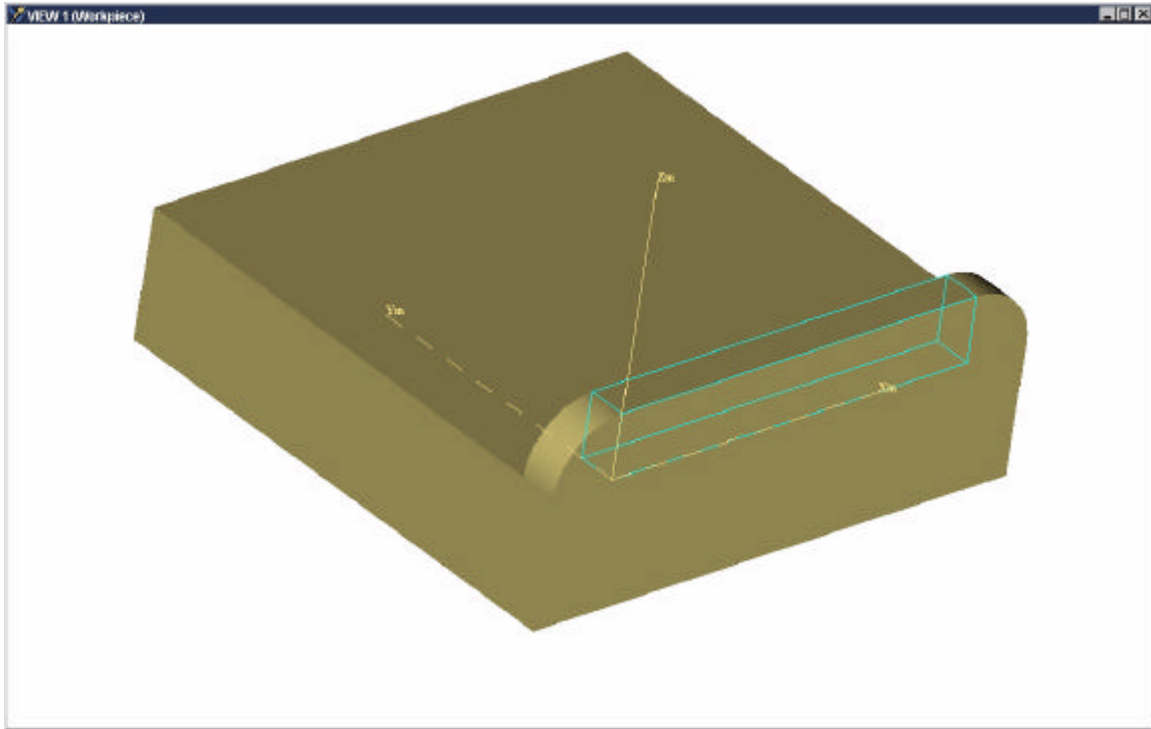
5. Add a block primitive to represent the standing web between the cylinders.

Click **Model** tab

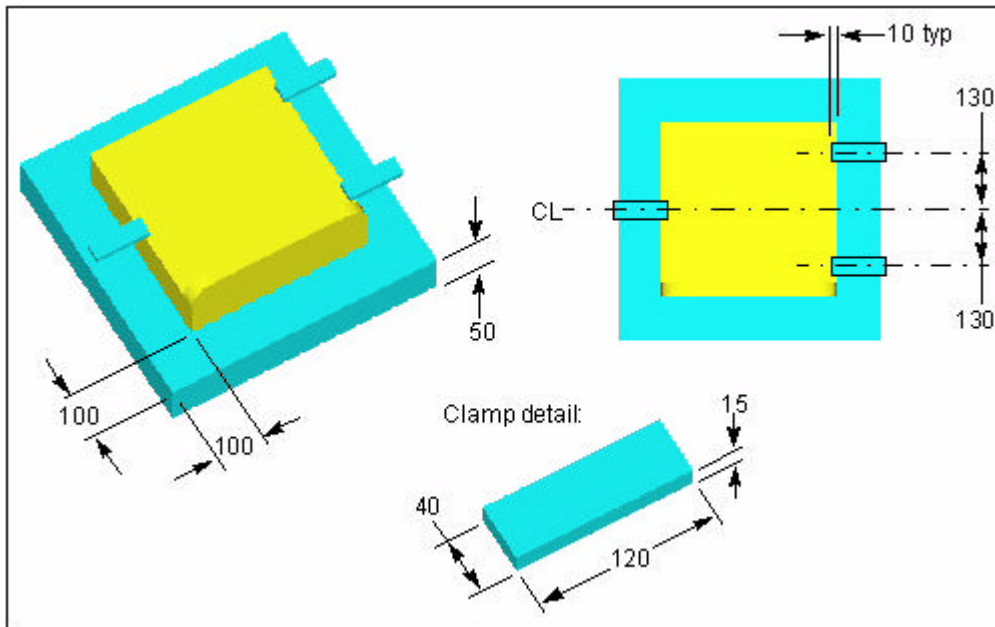
Type=Block, Length(X)=300, Width(Y)=30, Height(Z)=50, Add

Click **Position** tab

Position=50 0 100, Apply, OK



The 2D drawing below shows the fixture used to hold the casting for machining.



6. Add a fixture model base plate.

In the **Component Tree Window**, click on "Fixture(0,0,0)"

In the Modeling window click: **Model tab**

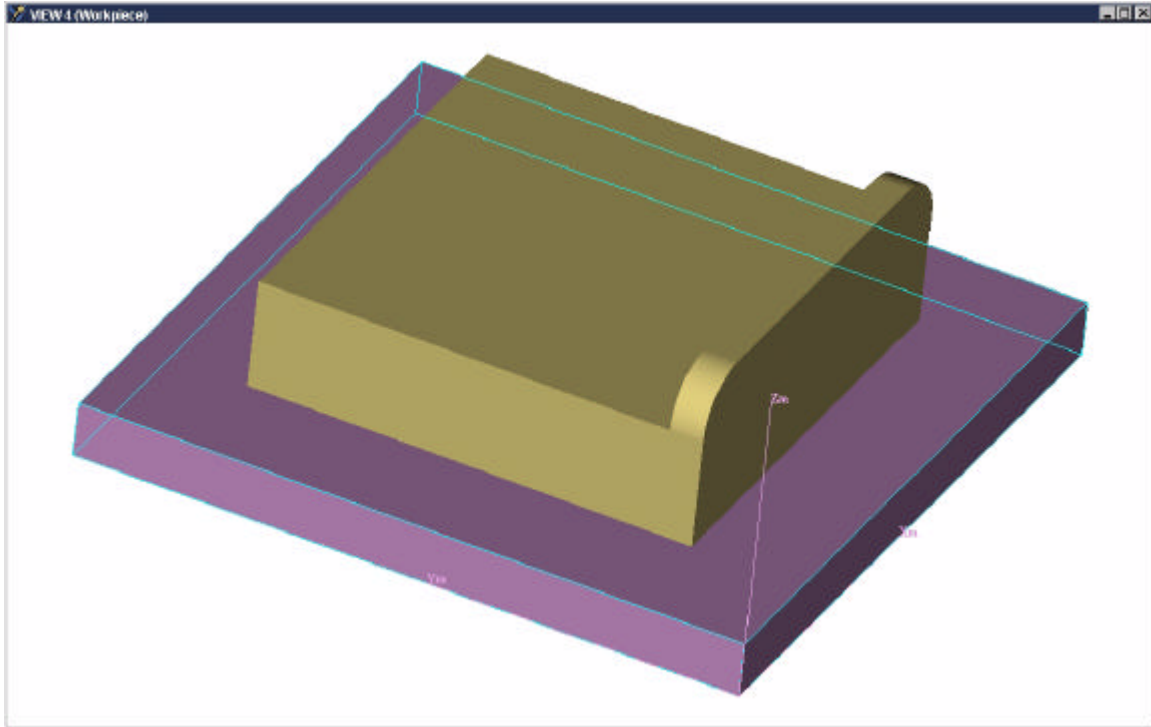
Type=Block, Length(X)=600, Width(Y)=600, Height(Z)=50, Add

Help on VERICUT

Click **Position** tab

Position= -100 -100 -50, **Apply**

Fit



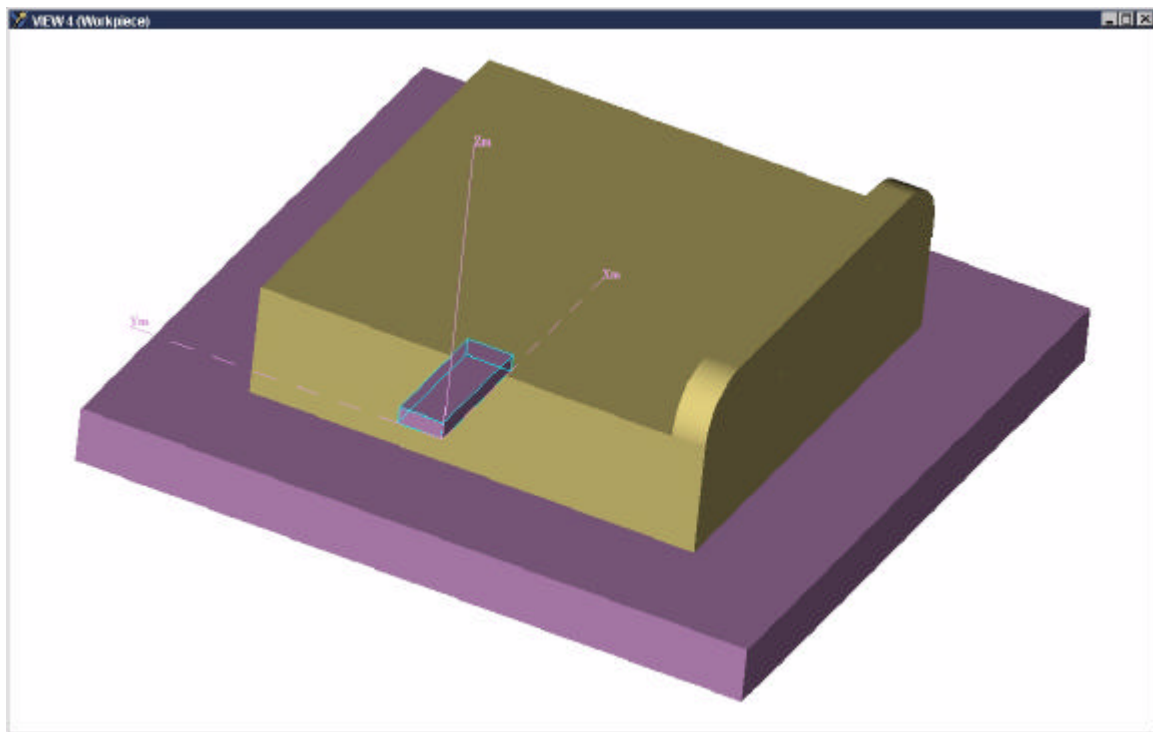
7. Add block models to represent the 3 clamps.

Click **Model** tab

Type=Block, Length(X)=120, Width(Y)=40, Height(Z)=15, Add

Click **Position** tab

Position= -110 180 100, **Apply**

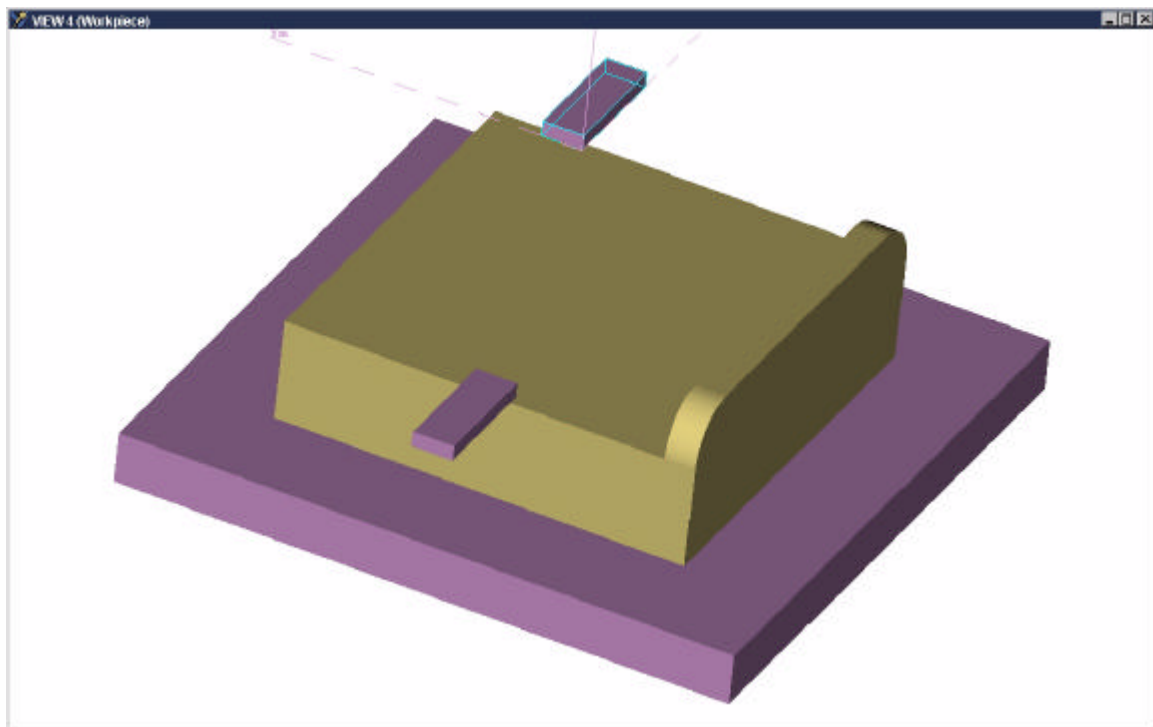


Click **Model** tab

Add

Click **Position** tab

Position=390 310 100, Apply



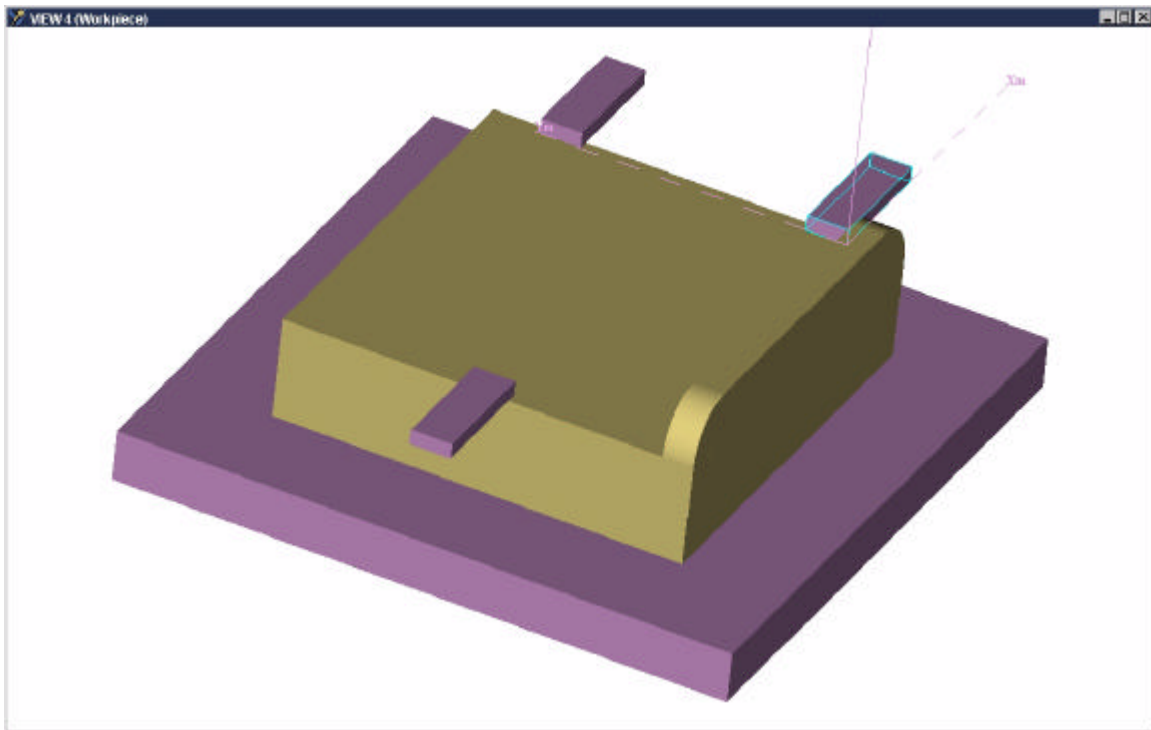
Help on VERICUT

Click **Model tab**

Add

Click **Position tab**

Position=390 50 100, Apply, OK



8. Exit the VERICUT session.

File > Exit, No

Session 3- Modeling and assembling models

This session shows how to use modeling features to assemble a stock model and a fixture. The **Assemble** features allow you to assemble the models by mating and aligning some of the model features. In both methods, the selected model will be moved so that the selected faces are co-planar. For **Align**, the normal vectors will be pointing in the same direction. For **Mate**, those vectors will be pointing in opposite directions.

Session Steps:

1. Start from a new Inch User file

File > Properties

Default units=Inch, OK

File > New

2. Display the model axis system

View > Axes

Select **Model**

Close

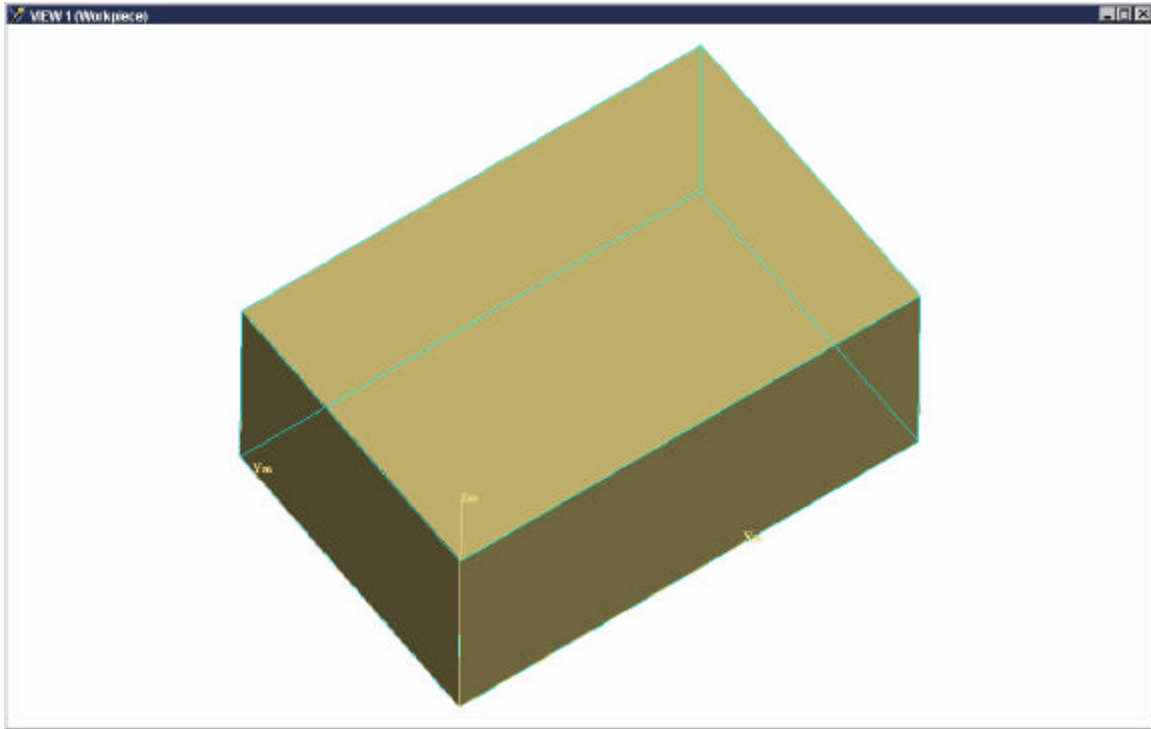
3. Add a Stock model to be a 9 x 6 x 4 inch block

Model > Model Definition: Model tab

Type=Block, Length(X)=9, Width(Y)=6, Height(Z)=4, Add

Fit





4. Add Fixture base model

Active Component=Fixture

Model tab

Type=Model File, Browse

Shortcut=CGTECH_LIBRARY

File Name=angloc_base.stl, OK

Add

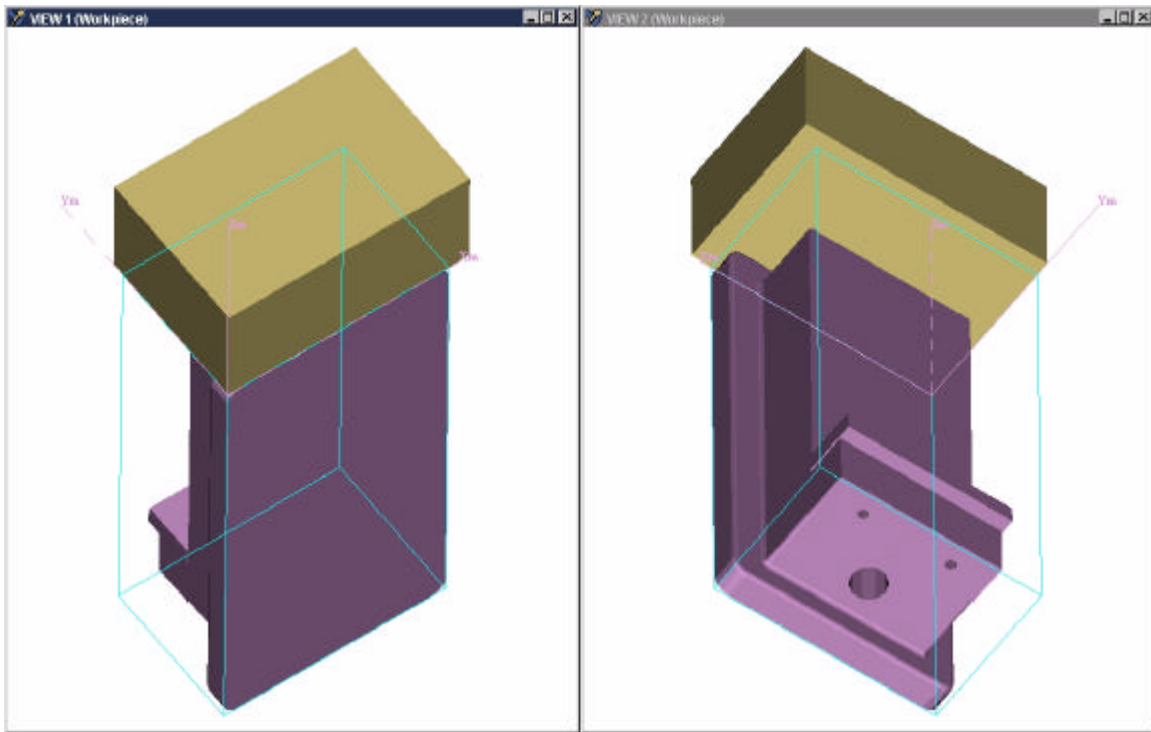
Fit 

View > Layout > Standard > click on 2 views (side by side)

View > Orient: Fit All

Left click in VIEW 2

Reverse 



5. Assemble the stock on the fixture

Model > Model Definition

Active Component=Stock (can also be selected by clicking on the model itself)

Assemble tab

F1, Help opens

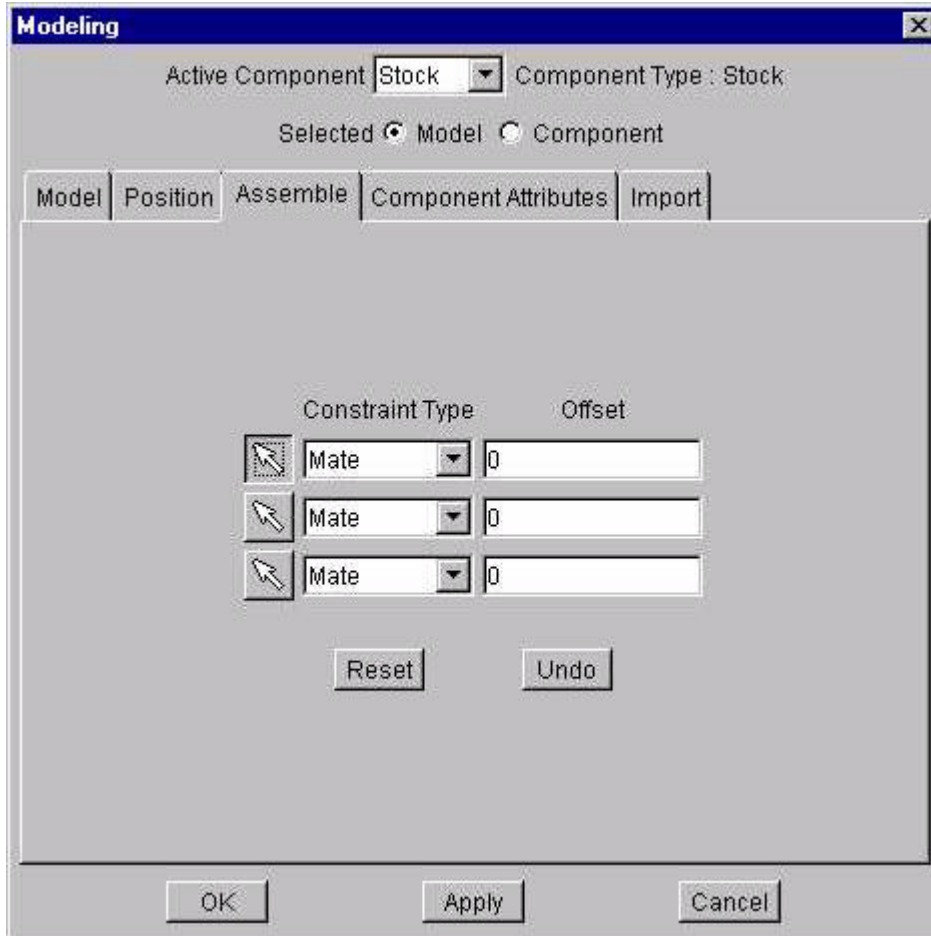
Click on **Assemble tab Features** for more information on assembling models

Close the web browser

Click on the arrow by the first **Constraint Type** 

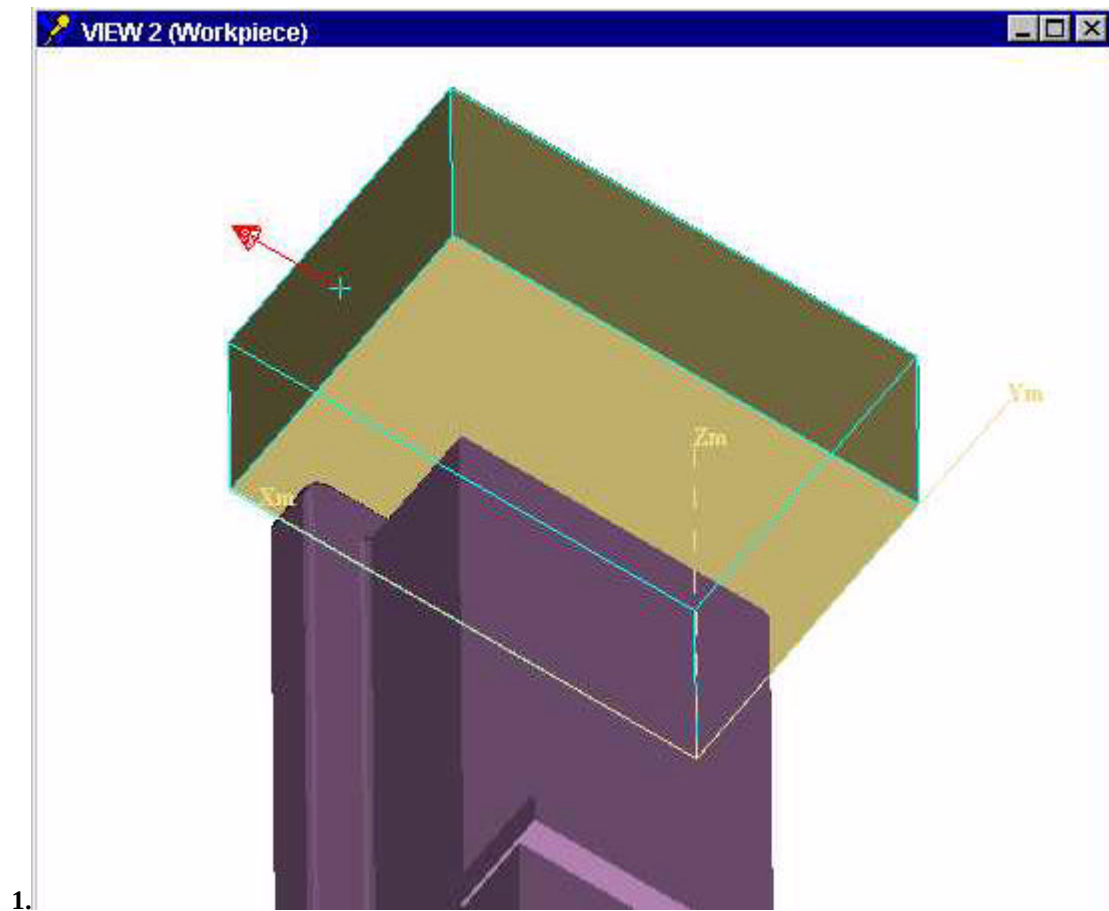
Constraint Type=Mate

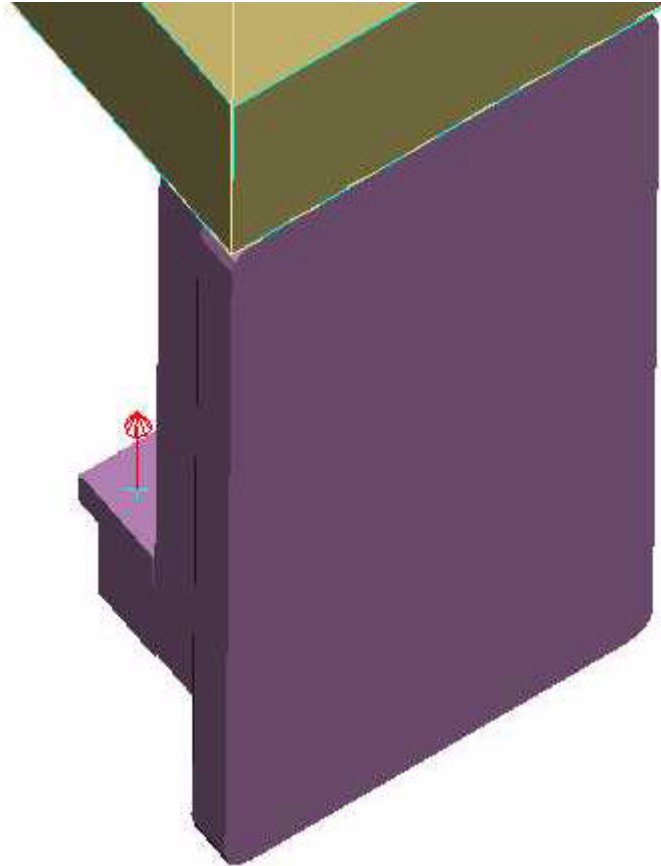
Note: Ensure that the Stock model is highlighted



Click on the stock face to mate (1)

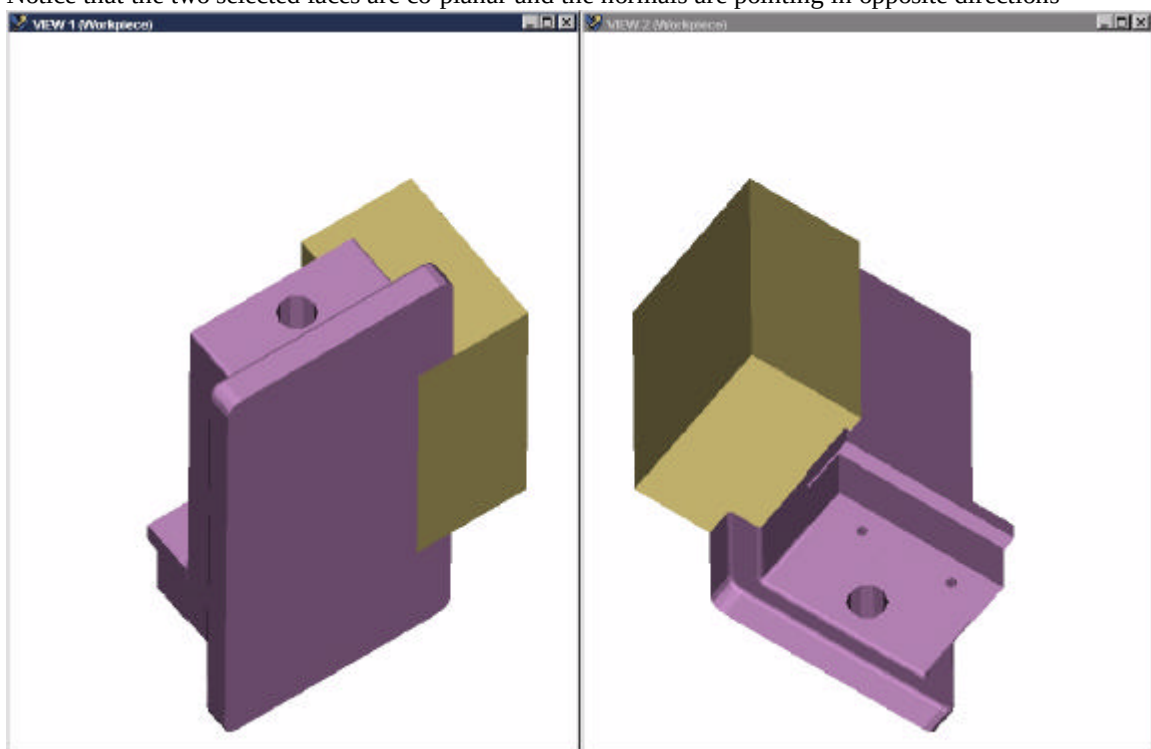
Click on the face of the jaw as shown (2)





2.

Notice that the two selected faces are co-planar and the normals are pointing in opposite directions

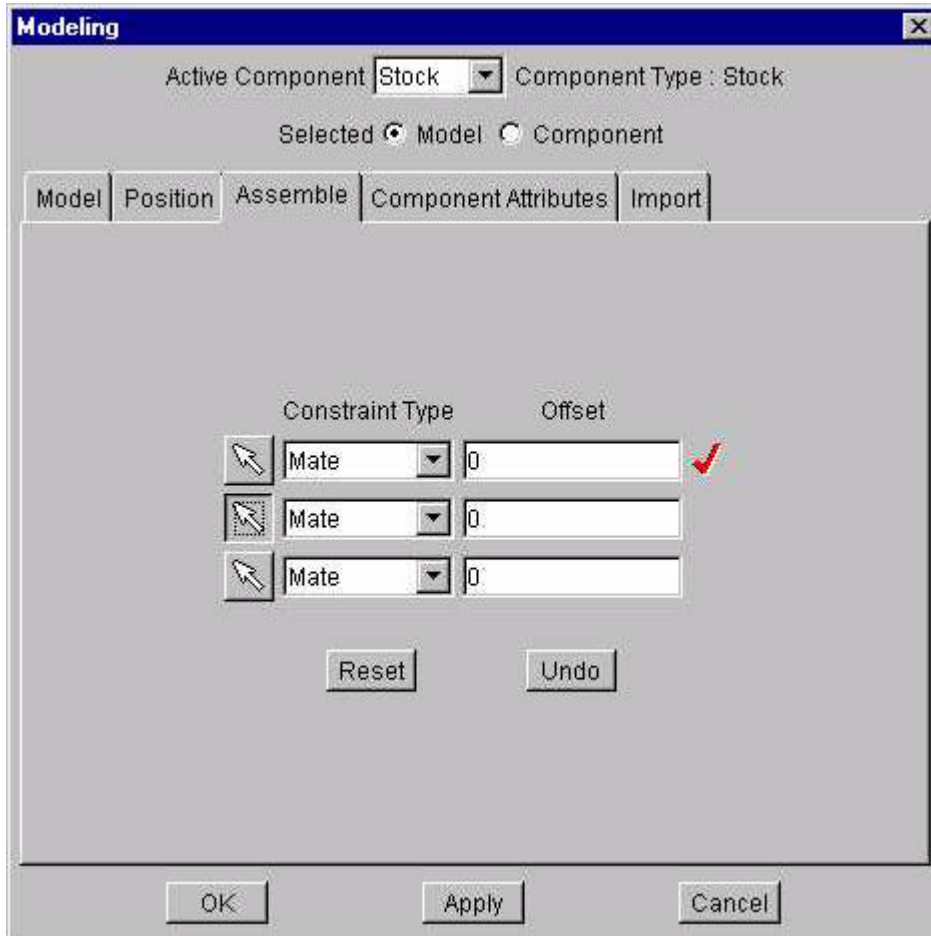


Click on the arrow by the Second **Constraint Type**



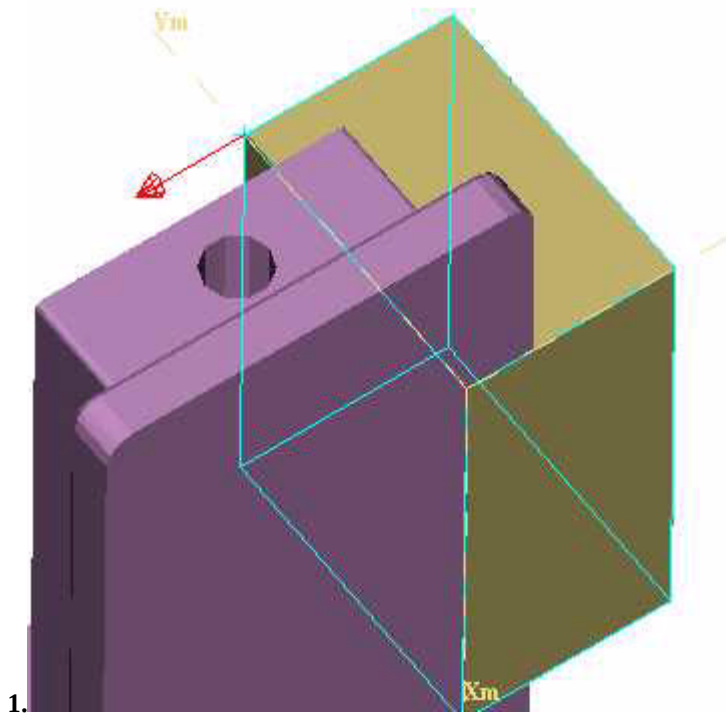
Constraint Type=Mate

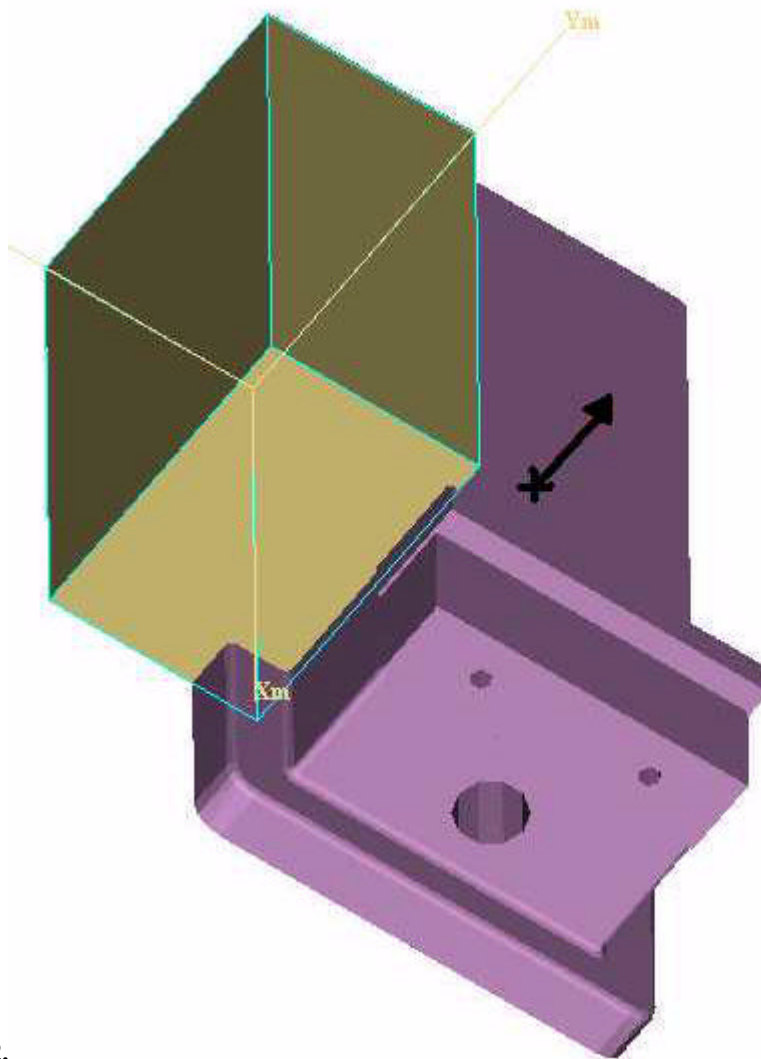
VERICUT shows the constraint is satisfied by highlighting it with a checkmark.



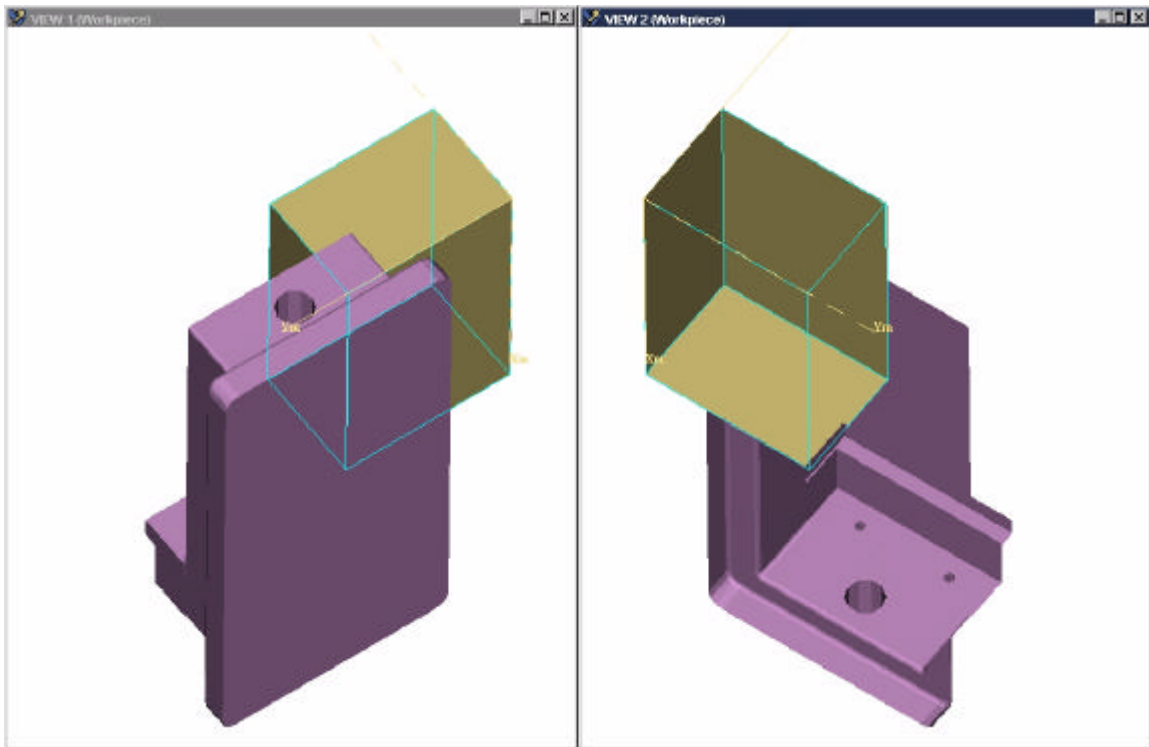
Click on the bottom stock model face to mate (1)

Click on the Fixture model (2)





2.

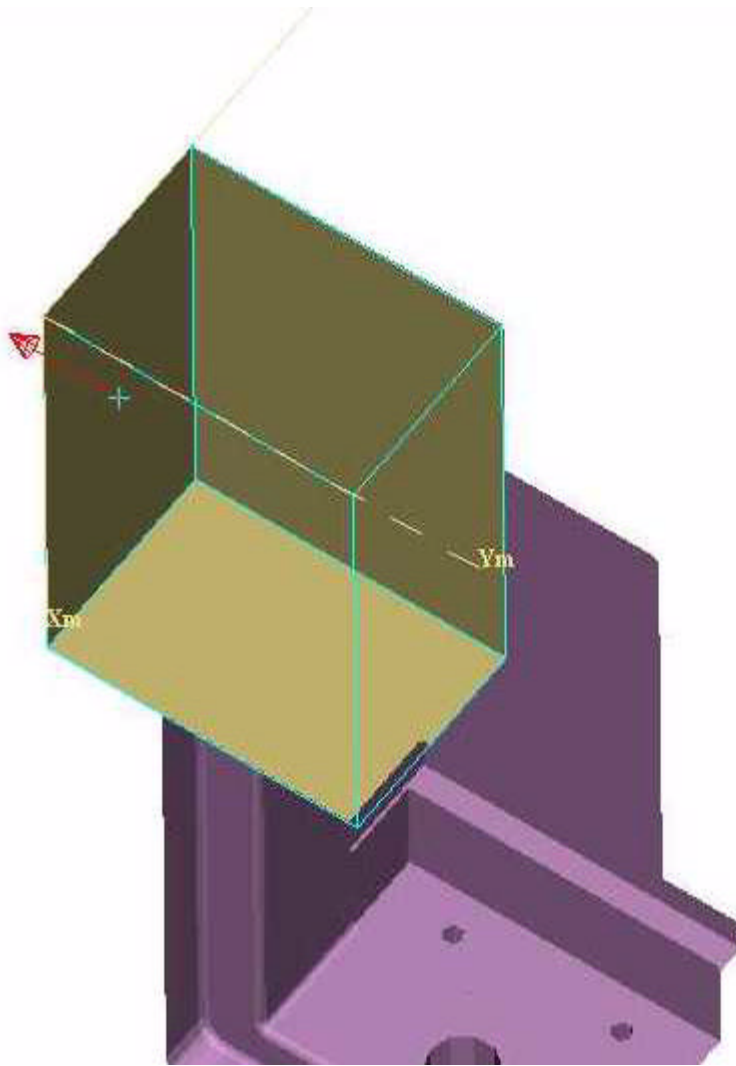


Click on the arrow by the Third **Constraint Type** 

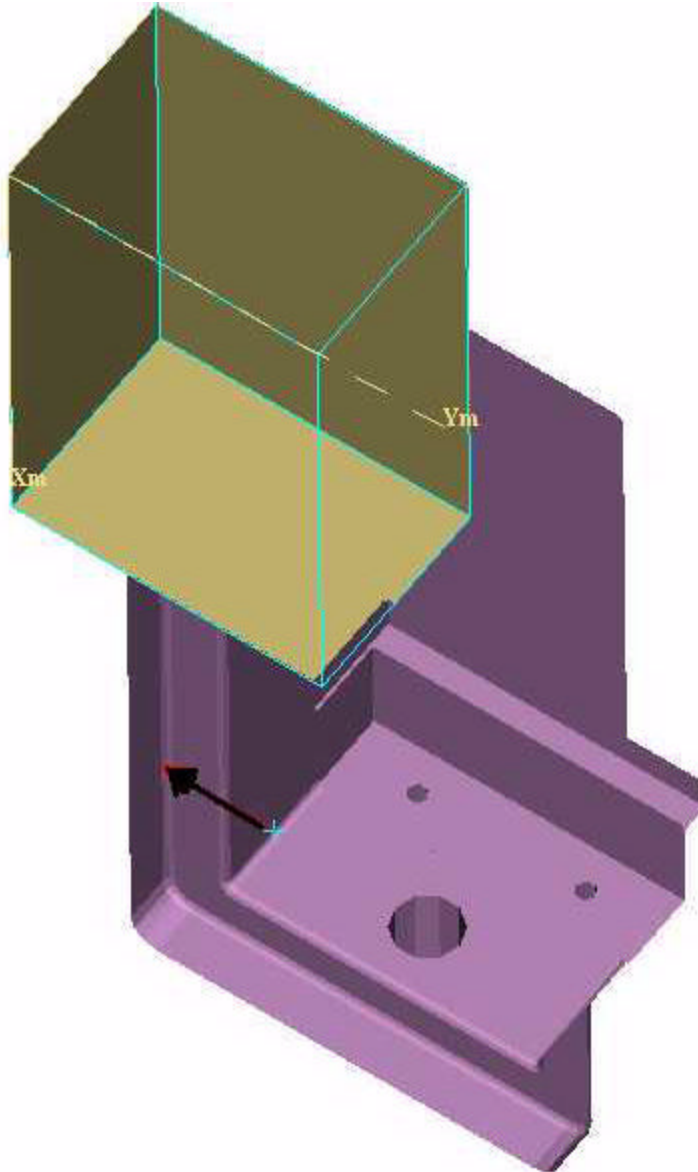
Constraint Type=Align

Click on the stock model side face to mate (1)

Click on the Fixture model (2)

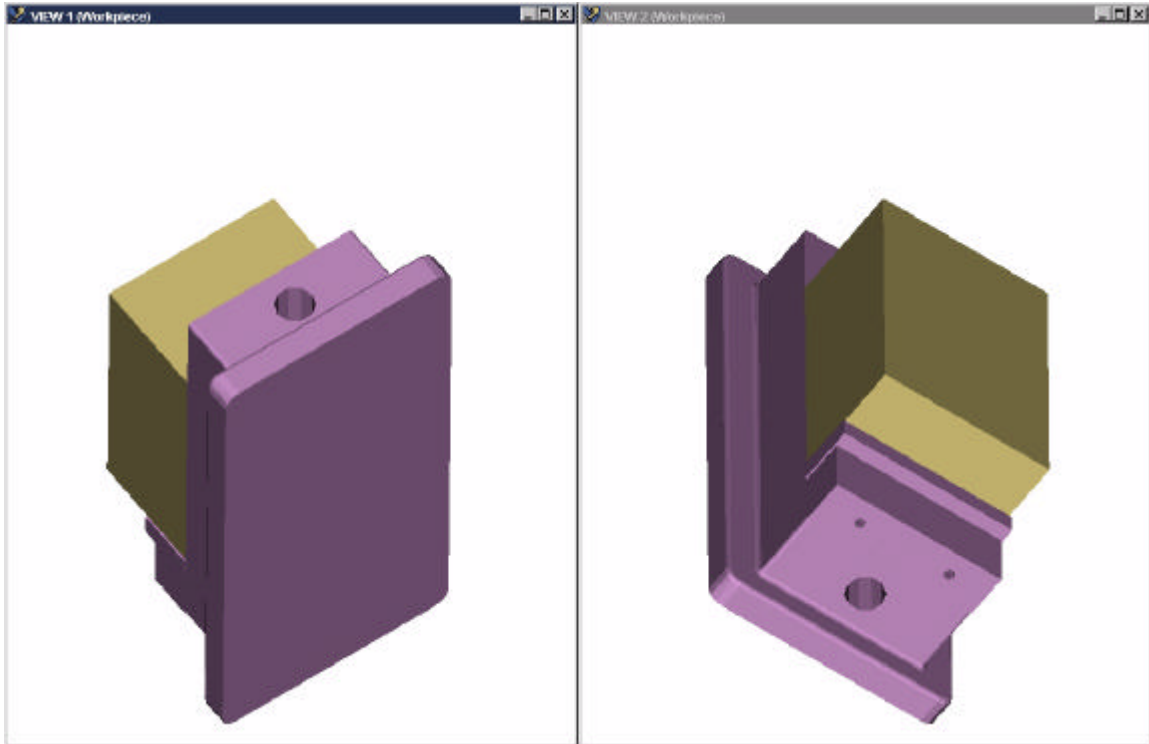


1.



2.

OK



6. Add a fixture model moveable jaw

Model > Model Definition

Active Component=Fixture

Model tab

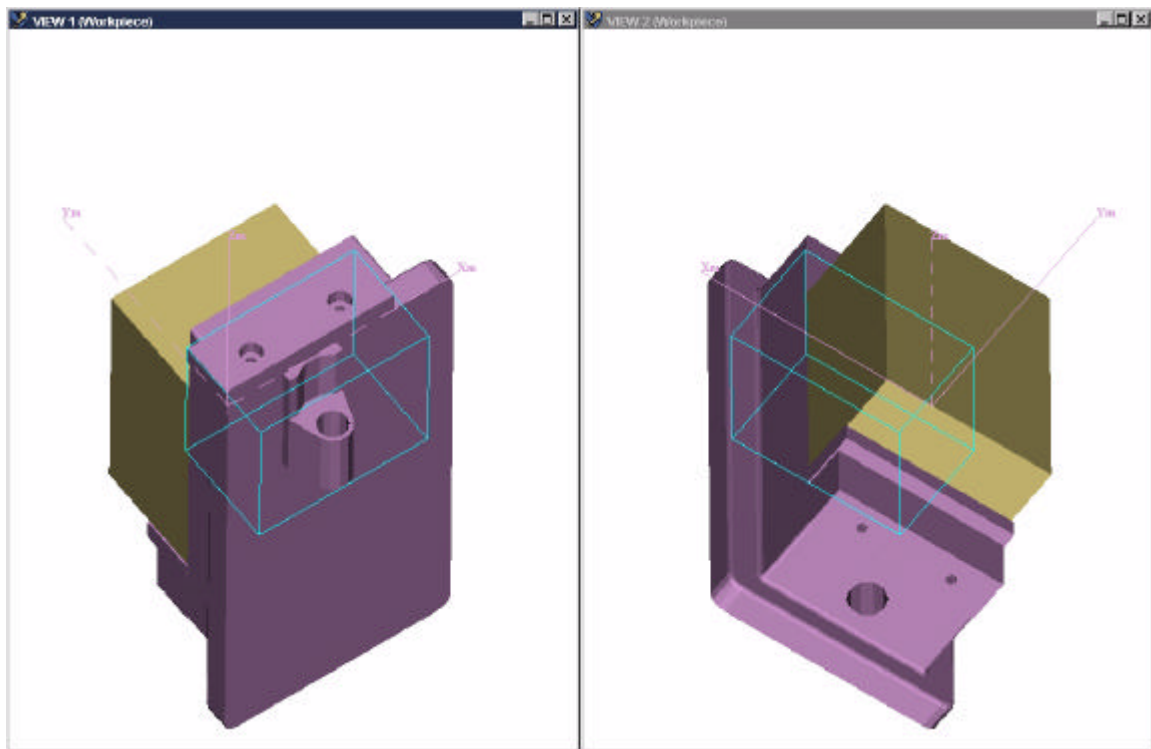
Type=Model File, Browse

Shortcut:CGTECH_LIBRARY

File Name=angloc_jaw.stl OK

Add

The jaw model is displayed inside the base



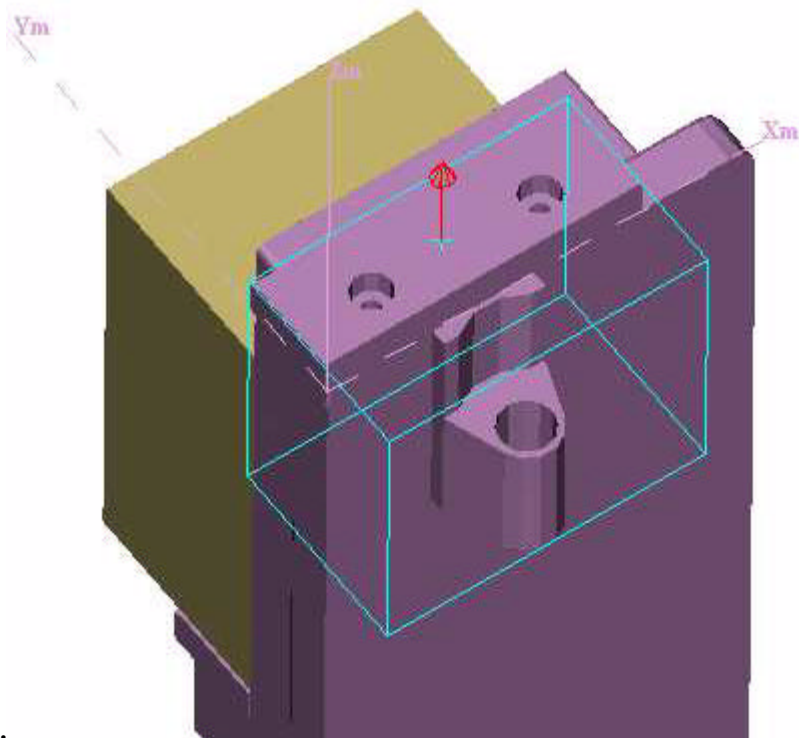
Assemble tab

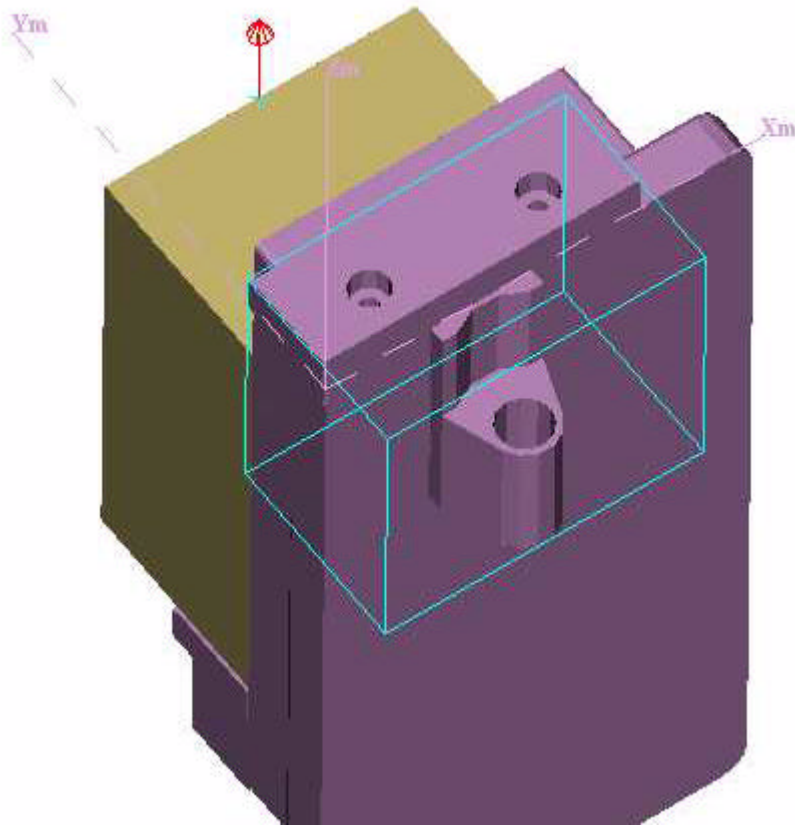
Click on the arrow by the first **Constraint Type** 

Constraint Type=Mate

Click on the face of the jaw model (1)

Click on the Stock model (2)





2.

Left click in **VIEW 1**

Reverse 

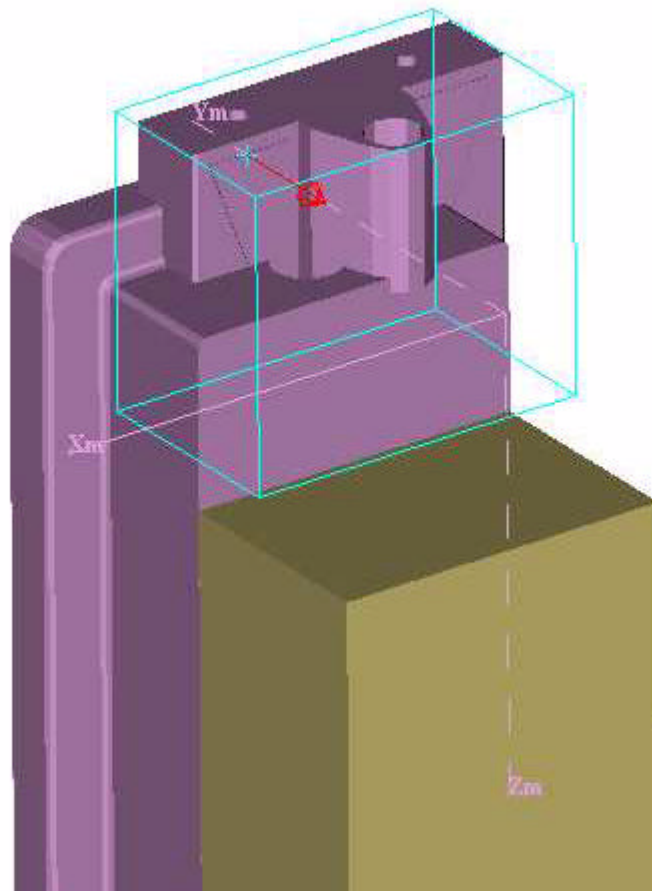
Dynamic X Rotation  - Drag the cursor downward to rotate the part until the jaw is available to selection

Click on the arrow by the Second **Constraint Type** 

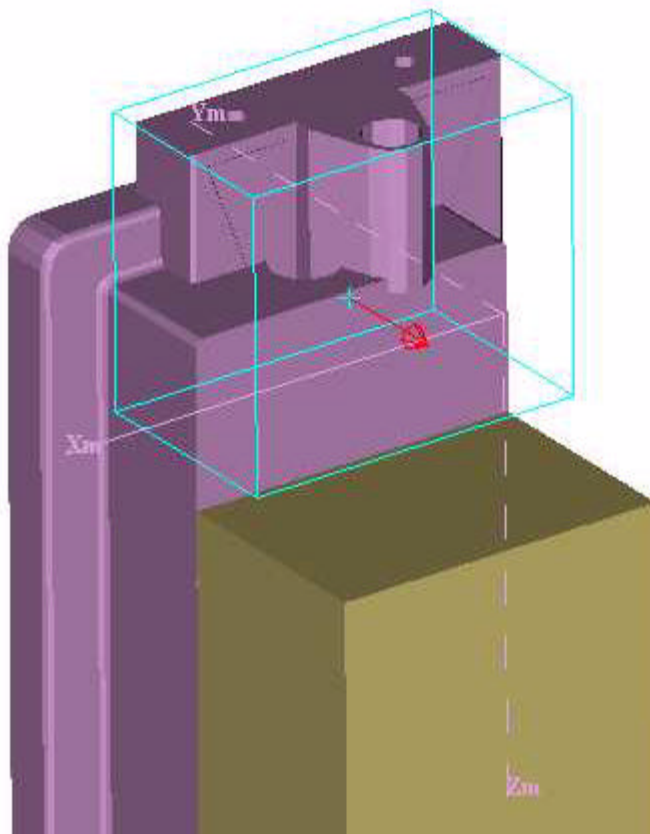
Constraint Type=Mate

Click on the jaw model bottom face face to mate (1)

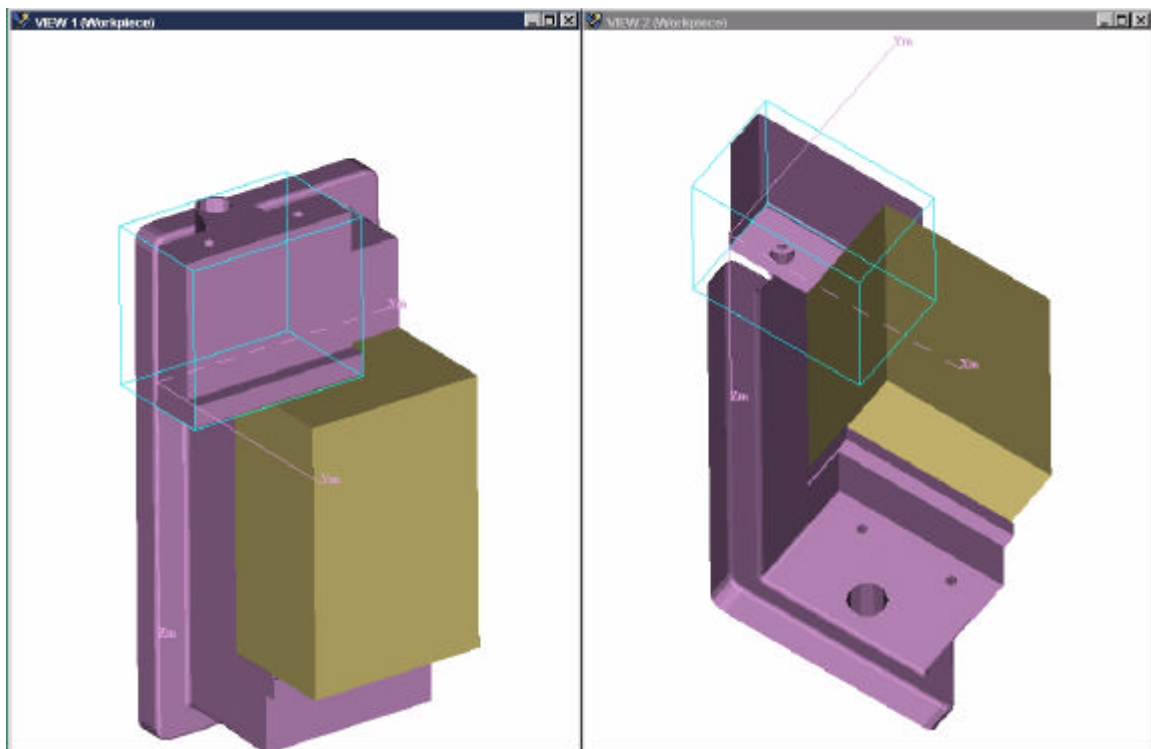
Click on the top of the base model (2)



1.



2.

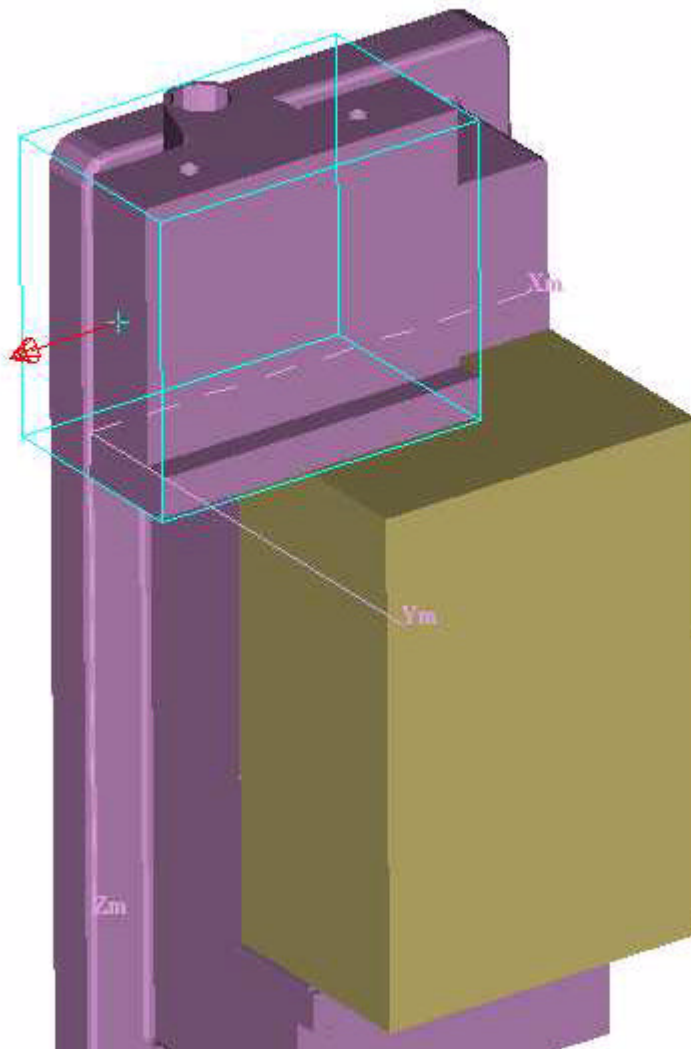


Click on the arrow by the third **Constraint Type** 

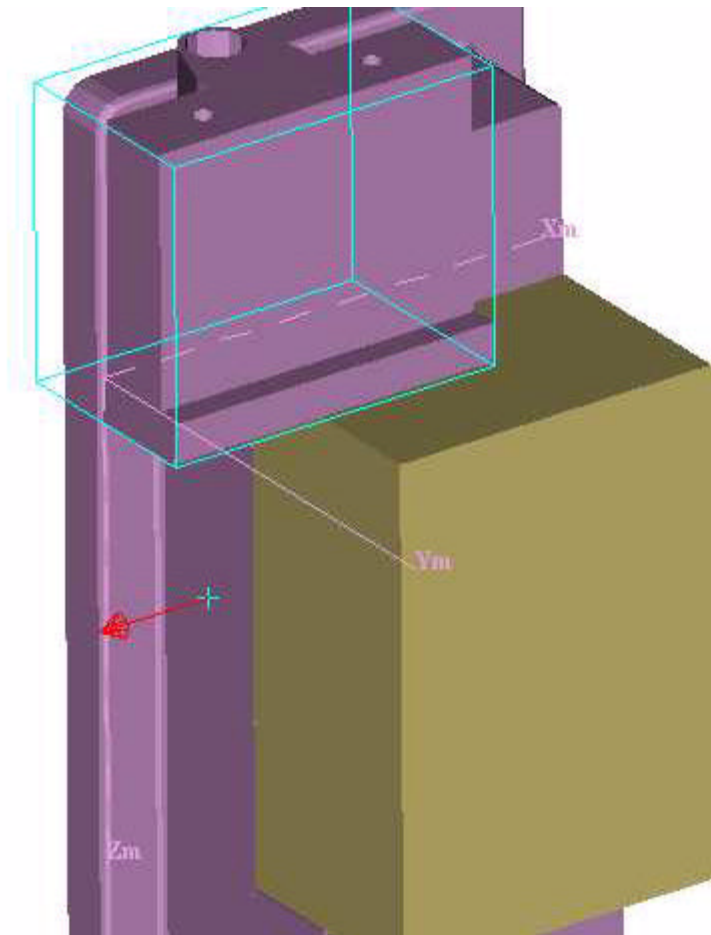
Constraint Type=Align

Click on the jaw model side face to mate (1)

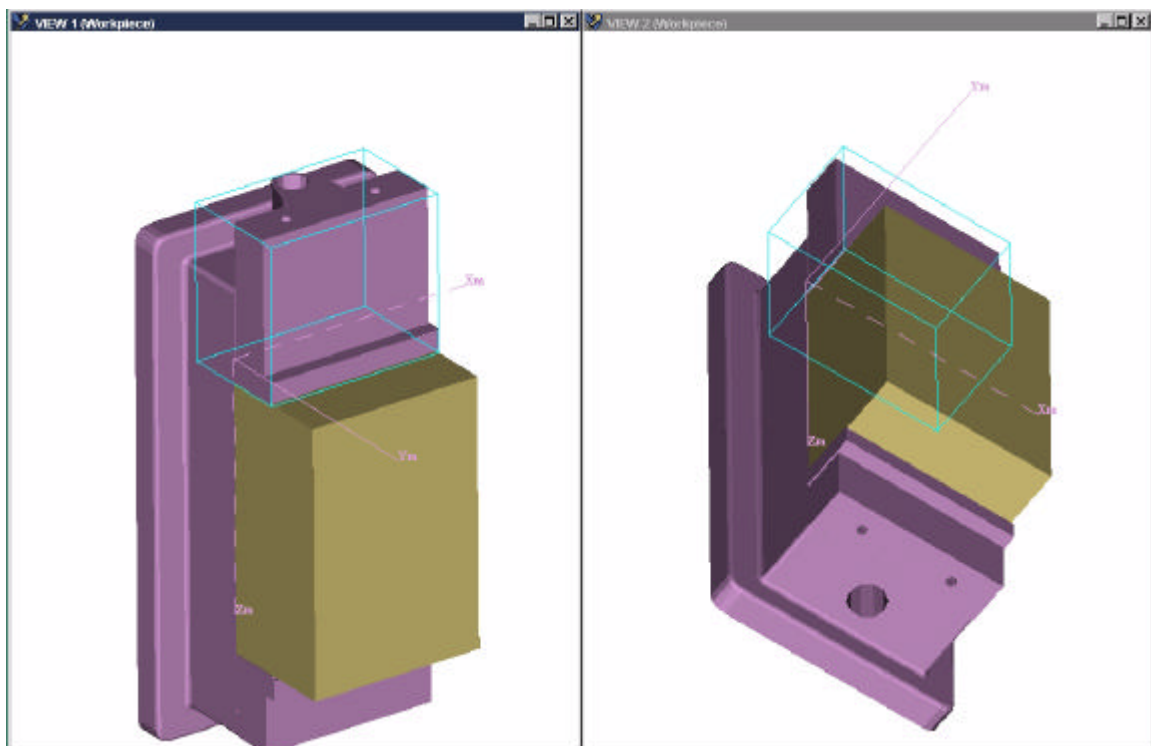
Click on the side of the base model (2)



1.



2.



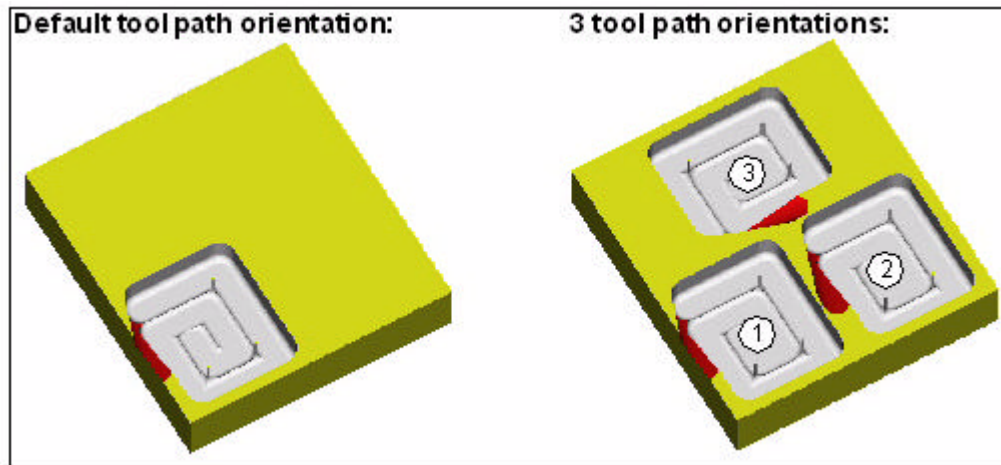
7. Exit the VERICUT session

File > Exit, No

Session 4- Cut multiple setups using the same tool path

This session shows how to use multiple work coordinates with a single tool path, such as required to simulate indexing a workpiece to different positions for cutting by the same tool path. The vcblock.tp tool path file is used with 3 different tool path orientations to machine 3 pockets in a 10 x 11 x 2 block of material (see below).

Pockets machined using "vcblock.tp":



Session Steps:

1. Start with a new Inch User file

File > Properties

Default units=Inch, OK

File > New

2. Add a Stock model to be a 10 x 11 x 2 inch block

Model > Model Definition: Model tab

Type=Block, Length(X)=10, Width(Y)=11, Height(Z)=2, Add

Fit 

3. Display the model axis system

View > Axes

Select **Model**

Select **Coordinate System**

Help on VERICUT

Close



4. Load the toolpath

Setup > Toolpath

Add

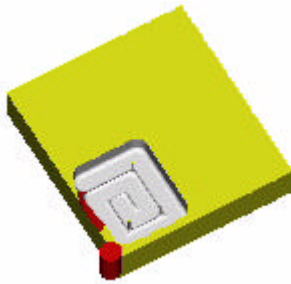
Shortcut=CGTECH_SAMPLES

File of type=*.tp

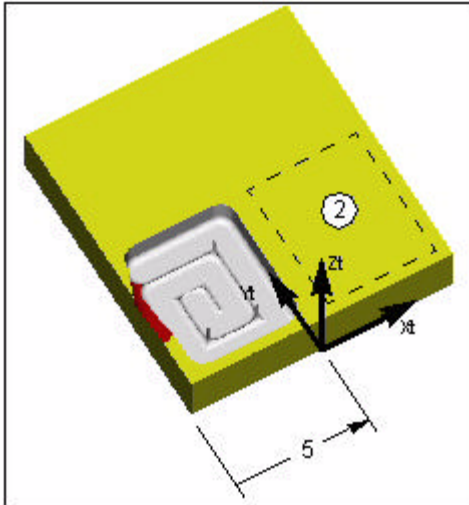
File Name=vcblock.tp, OK, OK

5. Cut the workpiece

Play to End 



Position 2 orientation (translated):



4. Modify the tool path orientation to cut pocket #2

Setup > Coordinate System

Coordinate System Name=orientation 1

Translate From=0 0 0, To=5 (or To=5 0 0), Move

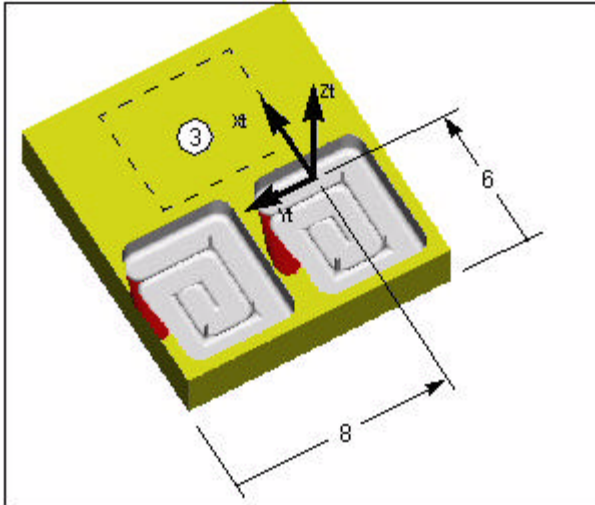
5. Rewind the tool path, then cut pocket #2

Rewind 

Play to End 



Position 3 orientation (rotated and translated):



6. Modify the tool path orientation to cut pocket #3

In **Coordinate System** window



Display the center of rotation, Click on

Center of Rotation=8 6 0, Apply

Position=8 6 0, Apply

Increment=90, Z+

7. Rewind the tool path and cut pocket #3

Rewind

Play to End



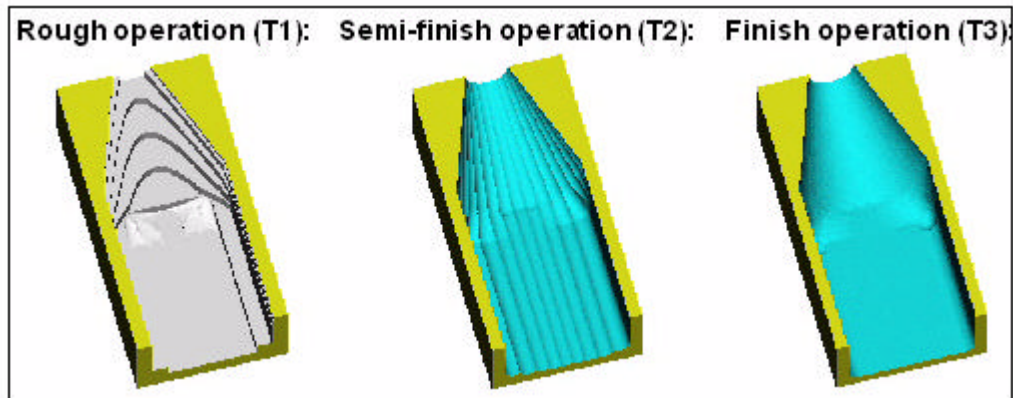
8. Exit the VERICUT session

File > Exit, No

Session 5- Simulate multiple tool path files, same orientation

This session shows how to process multiple tool paths during a single VERICUT session by configuring to simulate a series of tool paths which have the same orientation. However, VERICUT can also be easily configured simulate multiple tool paths having different orientations.

These metric tool paths machine the part in three stages: "bottle1.mcd" uses a single tool to rough machine a bottle mold cavity. The "bottle2.mcd" tool path uses a different tool to semi-finish the cavity, then "bottle3.mcd" finishes the cavity surfaces. All three tool paths use the same tool path orientation.



Session Steps:

1. In VERICUT, open the "gtooltip.usr" User file

File > Open

Shortcut=CGTECH_LIBRARY

File Name=gtooltim.usr, OK, OK (If a message appears, press **Reset.**)

3. Choose a 1 view layout

View > Layout > Standard > click on 1 view

4. Display the model axis system

View > Axes

Select **Model**

Select **Coordinate System**

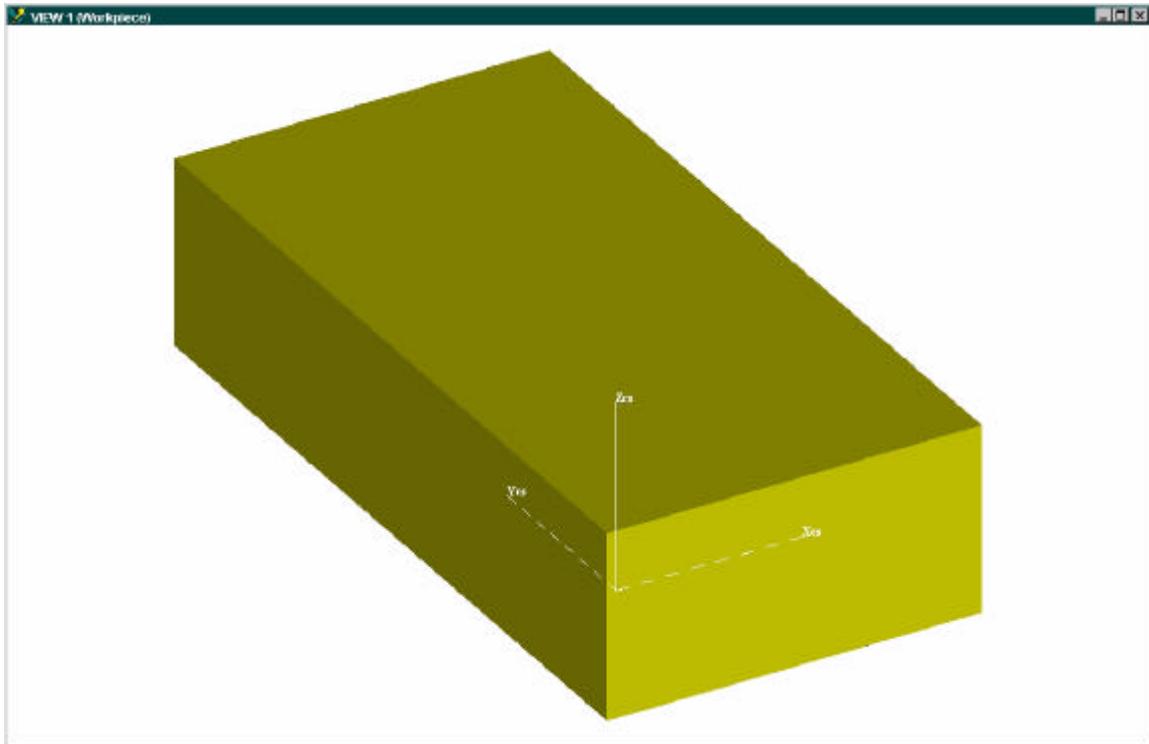
Close

5. Modify Stock model to be a 100 x 200 x 50 millimeter block

Model > Model Definition: Model tab

Type=Block, Length(X)=100, Width(Y)=200, Height(Z)=50, Apply

Fit 



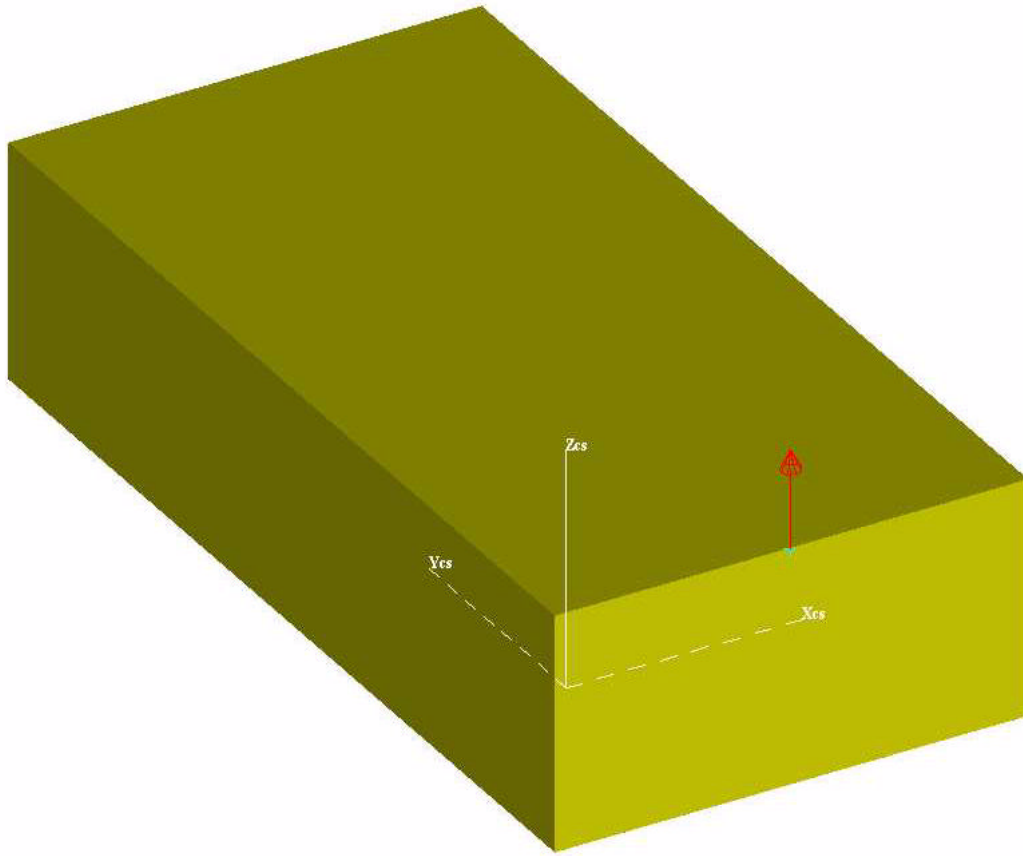
6. Modify the "orientation1" Coordinate System.

Setup > Coordinate System

Coordinate System Name=orientation 1

Click in the **Position** field (should become highlighted yellow)

Click on the top bottom edge center vector



Apply

7. Create a list of tool paths including "bottle1-3.mcd"

Setup > Toolpath

Select "mcdmillm.mcd"

Replace

Shortcut=CGTECH_SAMPLES

Using the **SHIFT** key, select: bottle1.mcd, bottle2.mcd, bottle3.mcd

OK, OK

8. Load the tool library "bottle.tls"

Setup > Tool Manager

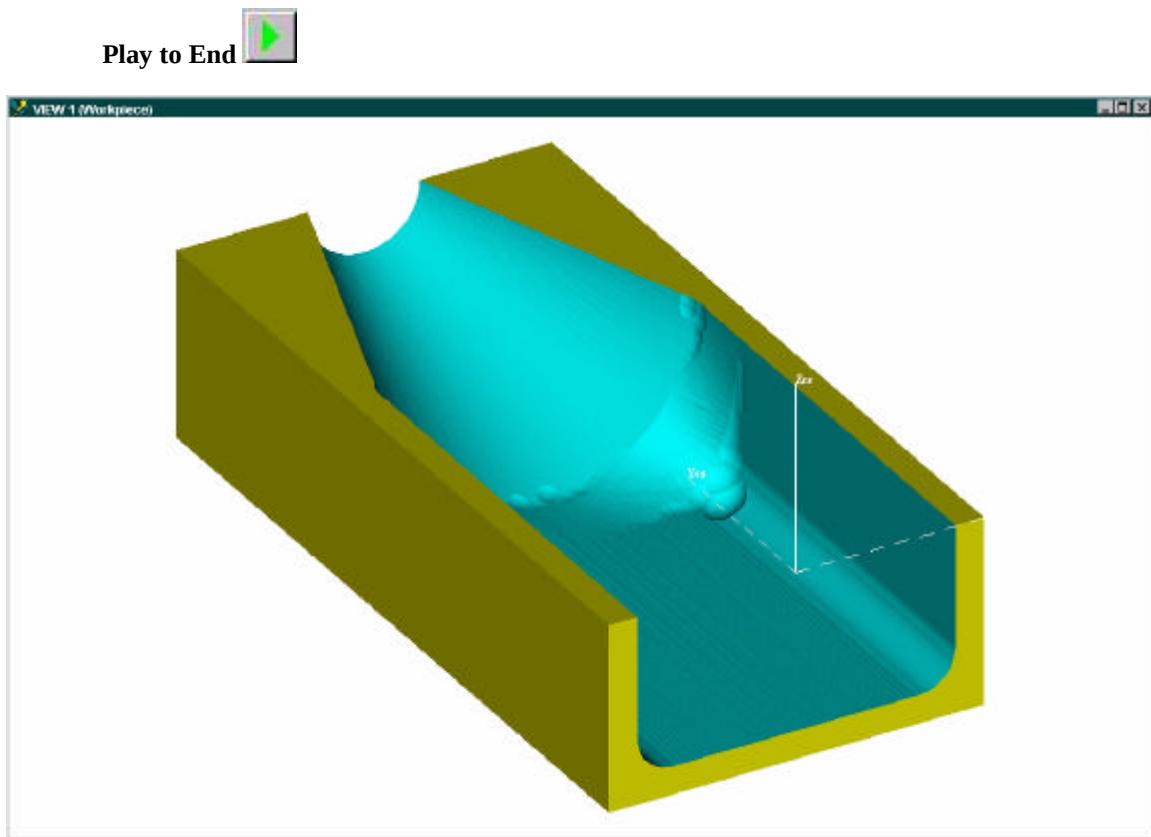
File > Open

Shortcut=CGTECH_SAMPLES

File Name=bottle.tls, Open

File > Close, Yes (to have VERICUT use the new Tool Library)

9. Simulate all three tool paths cutting the part



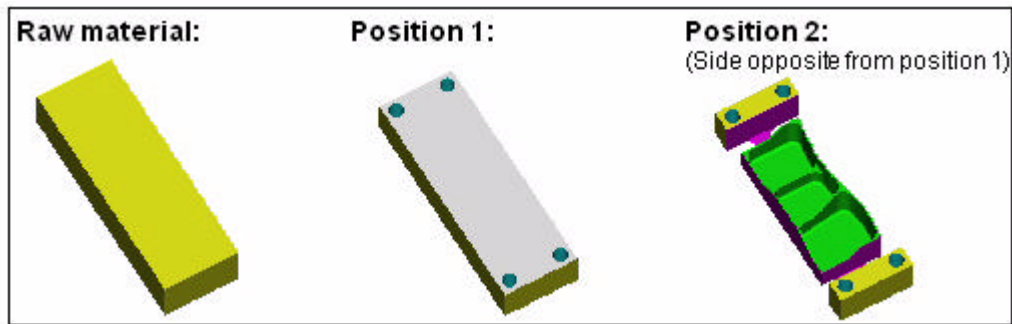
10. Exit the VERICUT session

File > Exit, No

Session 6- Simulate multiple tool paths, different orientations

This session shows how to assign multiple coordinate system orientations to multiple tool paths. User actions are similar to those demonstrated in **Cut Multiple Setups Using the Same Toolpath**, except a new tool path is specified for cutting with a new orientation.

The "flange1.tp" tool path machines the top of the part (fly-cut top surface and drill 4 holes). The "flange2.tp" tool path cuts 3 pockets and the profile on the position 2 side. These metric unit tool paths require a 100mm x 300mm x 60mm stock block.



Session Steps:

1. Start with a new millimeter User file

File > Properties

Default units=Millimeter, OK

File > New

2. Display the model axis system

View > Axes

Select **Model**

Select **Coordinate System**

Close

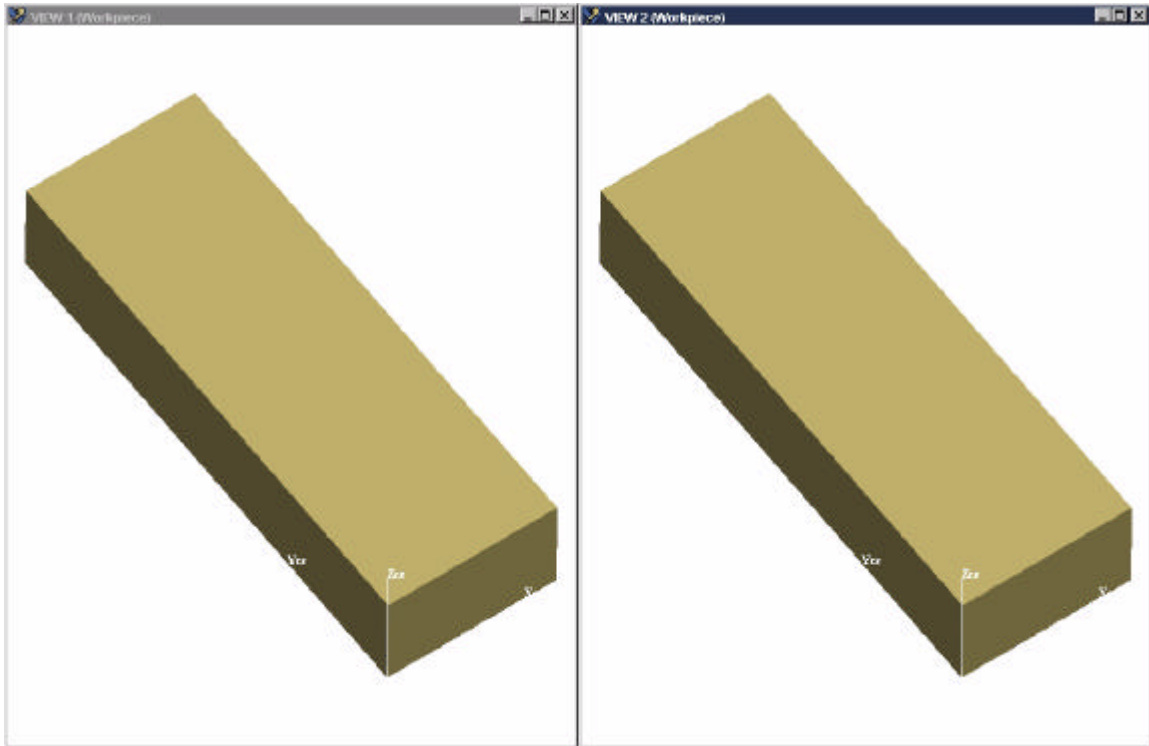
3. Choose a 2 view "side by side"

View > Layout > Standard > click on 2 views

4. Add a Stock model to be a 100 x 300 x 60 mm block

Model > Model Definition: Model tab

Type=Block, Length(X)=100, Width(Y)=300, Height(Z)=60, Add



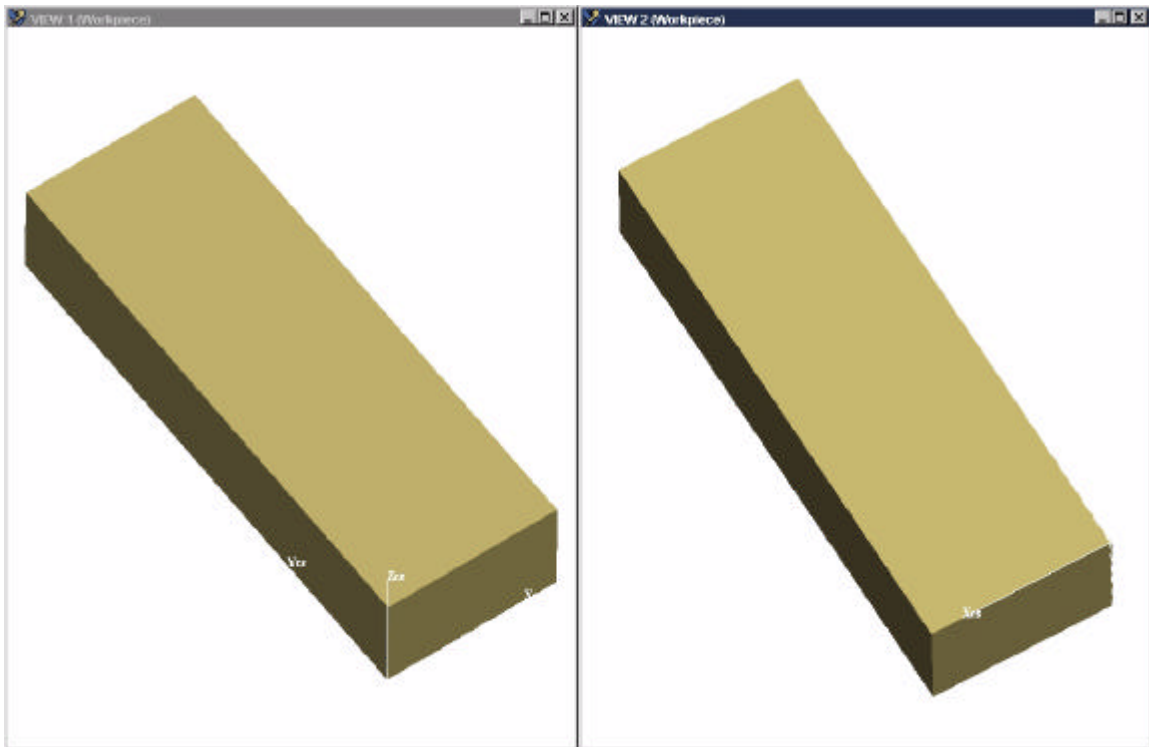
5. In the right side view use rotation angles "-153 -14 -153" to view the position 2 side, then re-fit the new view

Left click in **VIEW 2** to make it the active view

View > Orient

Angles=-153 -14 -153,Apply





5. Load the toolpath "flange1.tp"

Setup > Toolpath

Add

Shortcut=CGTECH_SAMPLES

File of type=*.tp

File Name=flange1.tp, OK, OK

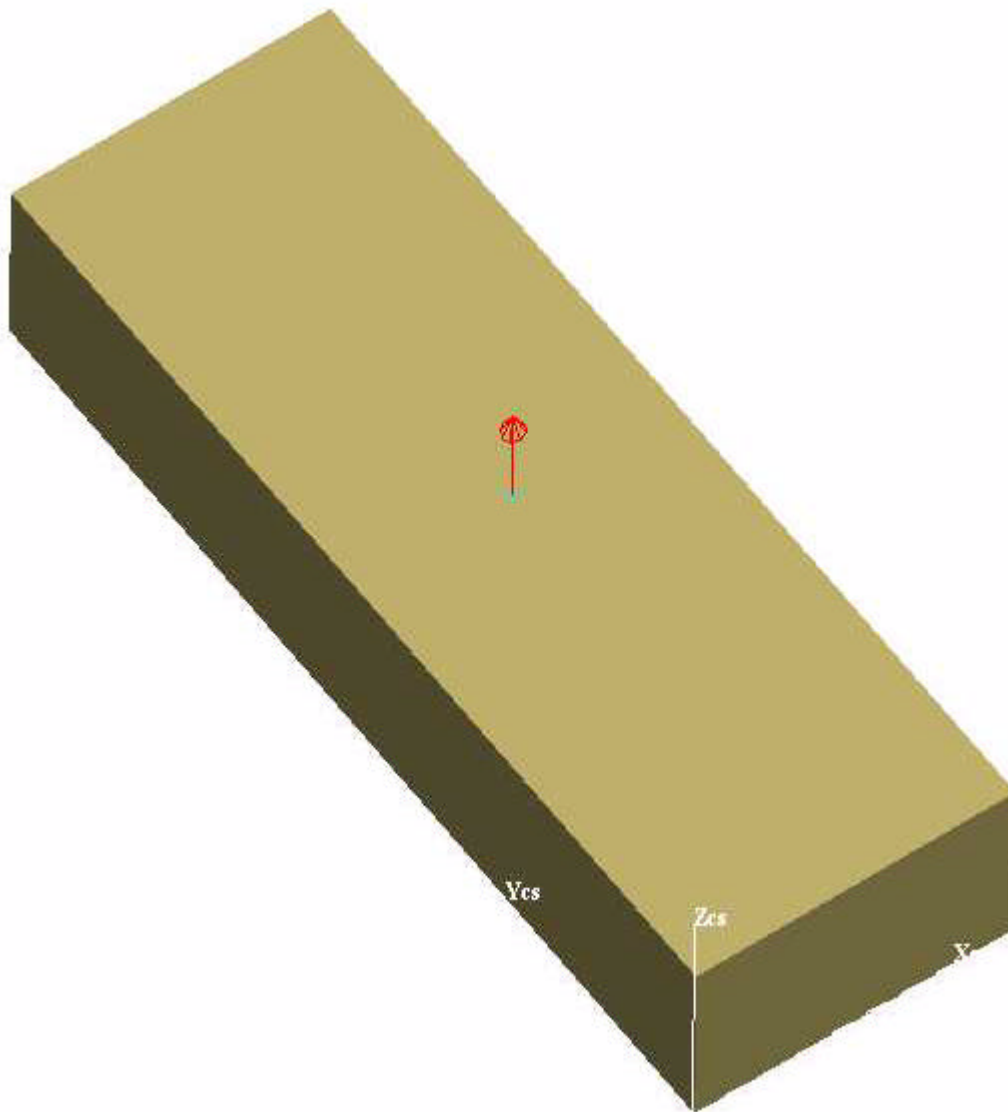
6. Orient "flange1.tp" to be on the top center of the stock block

Setup > Coordinate System

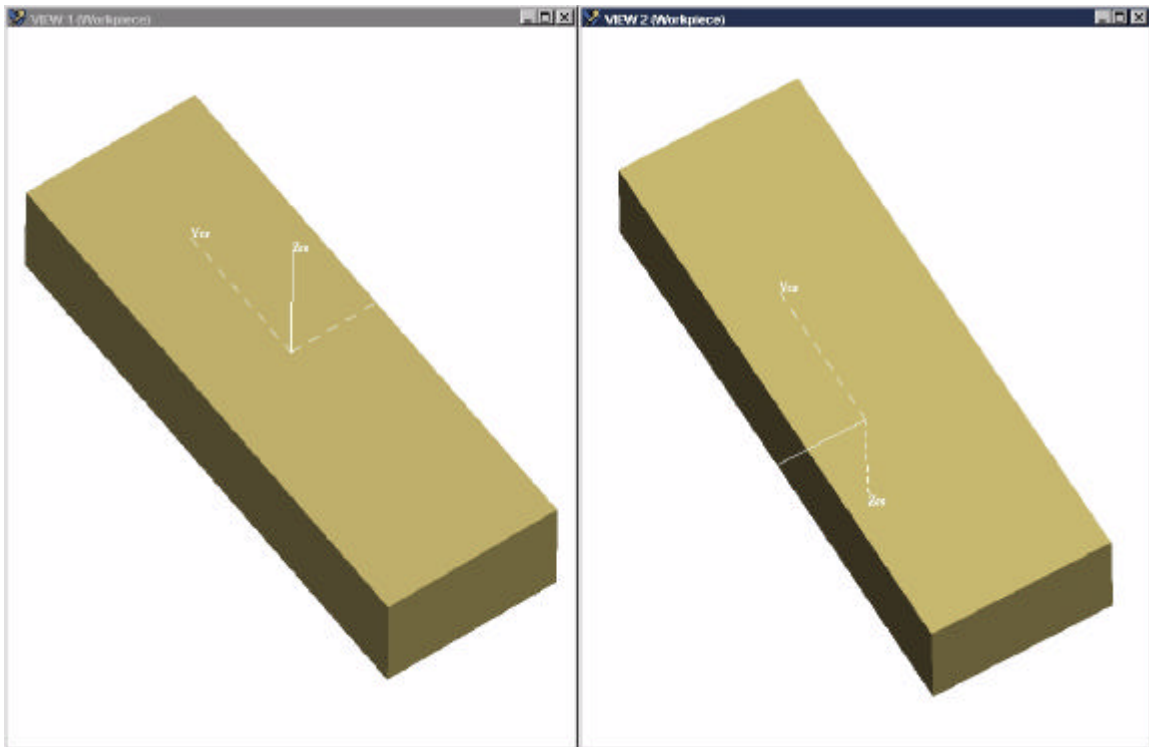
Coordinate System Name=orientation 1

Click in the **Position** field (should become highlighted yellow)

Click on the top center vector

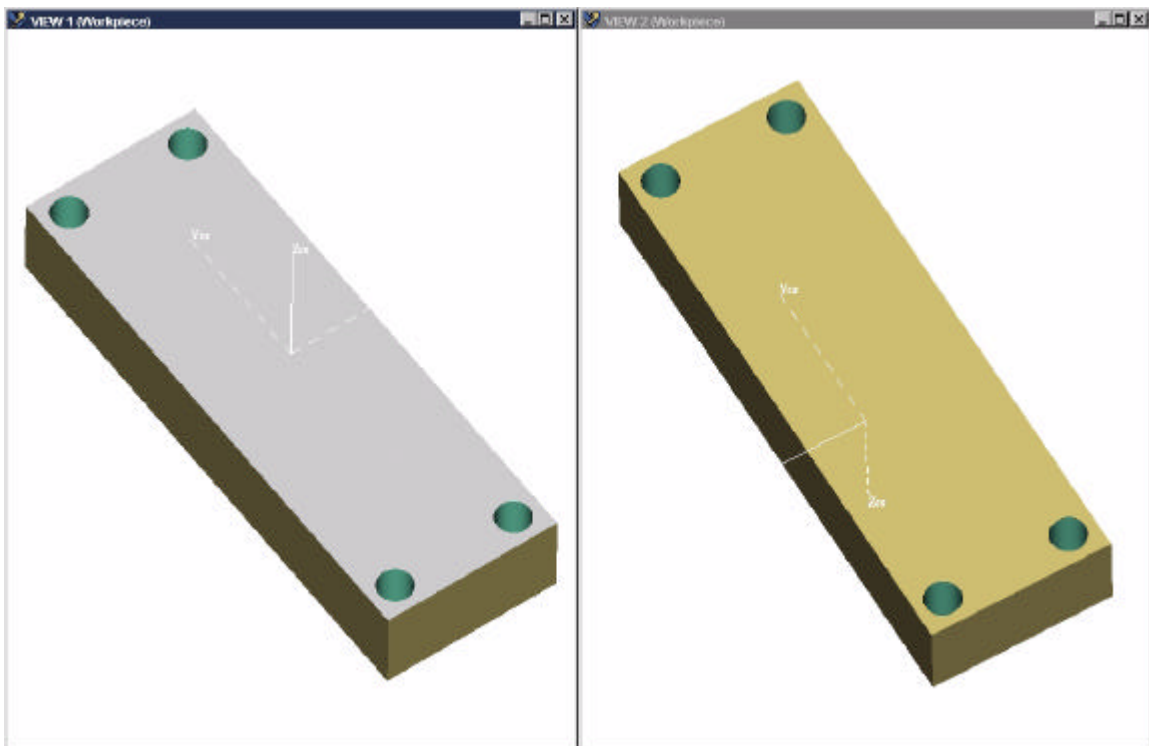


Apply

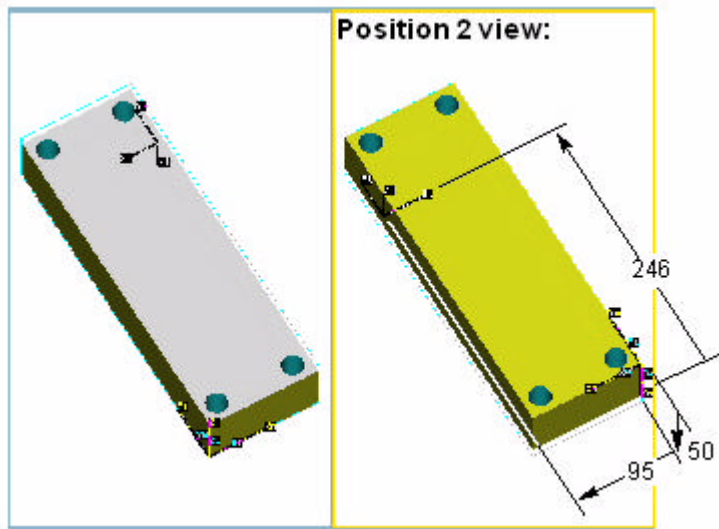


7. Cut the part in position 1

Play to End 



8. Orient "flange2.tp" to cut position 2



9. Modify the tool path orientation

Setup > Coordinate System

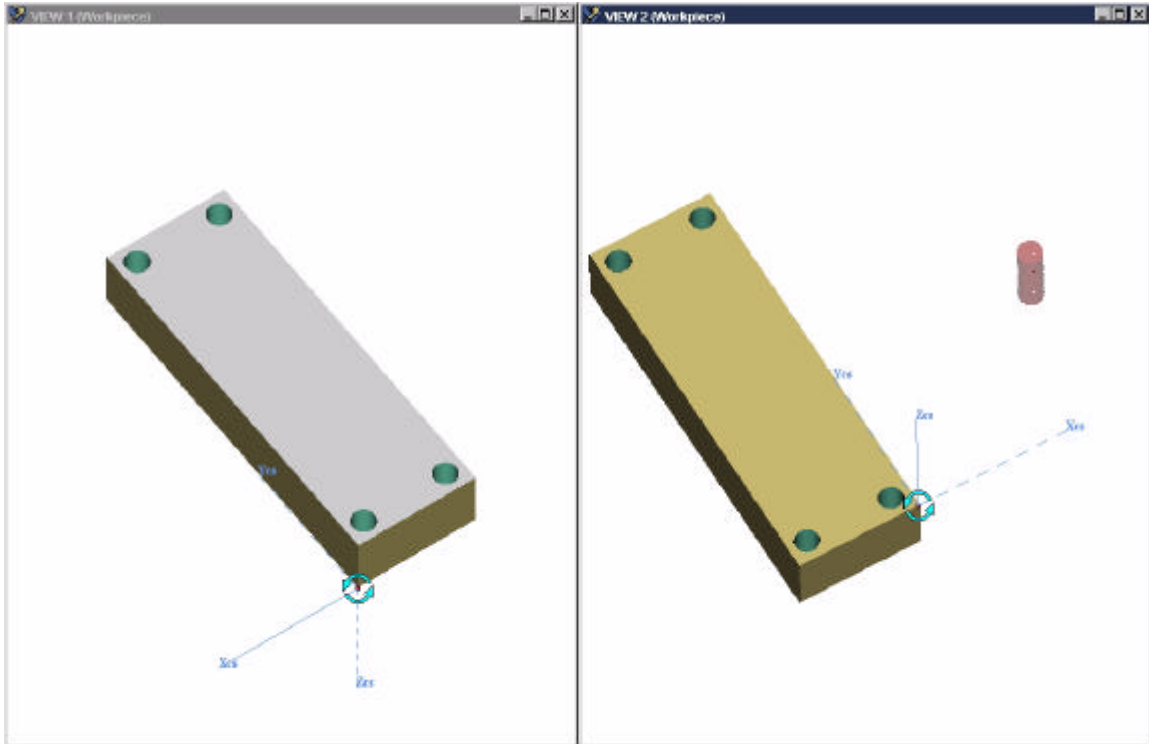
Coordinate System Name=Default

Coordinate System Name=orientation 2

Display the center of rotation, click on



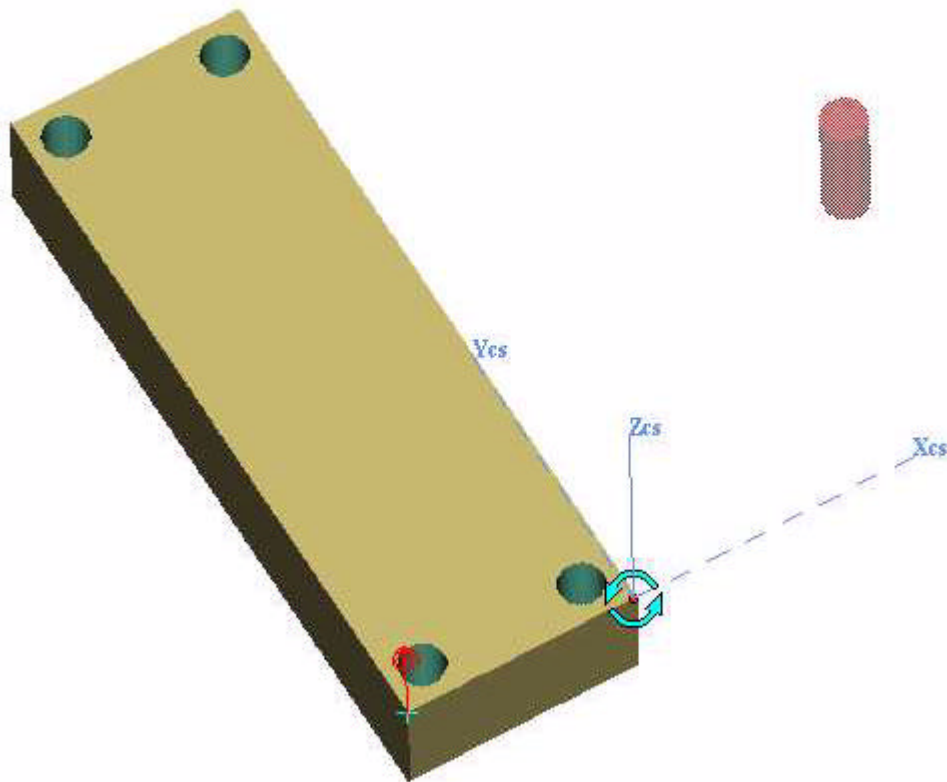
Increment=180, Y+



Left click in **VIEW 2** to make it the active view

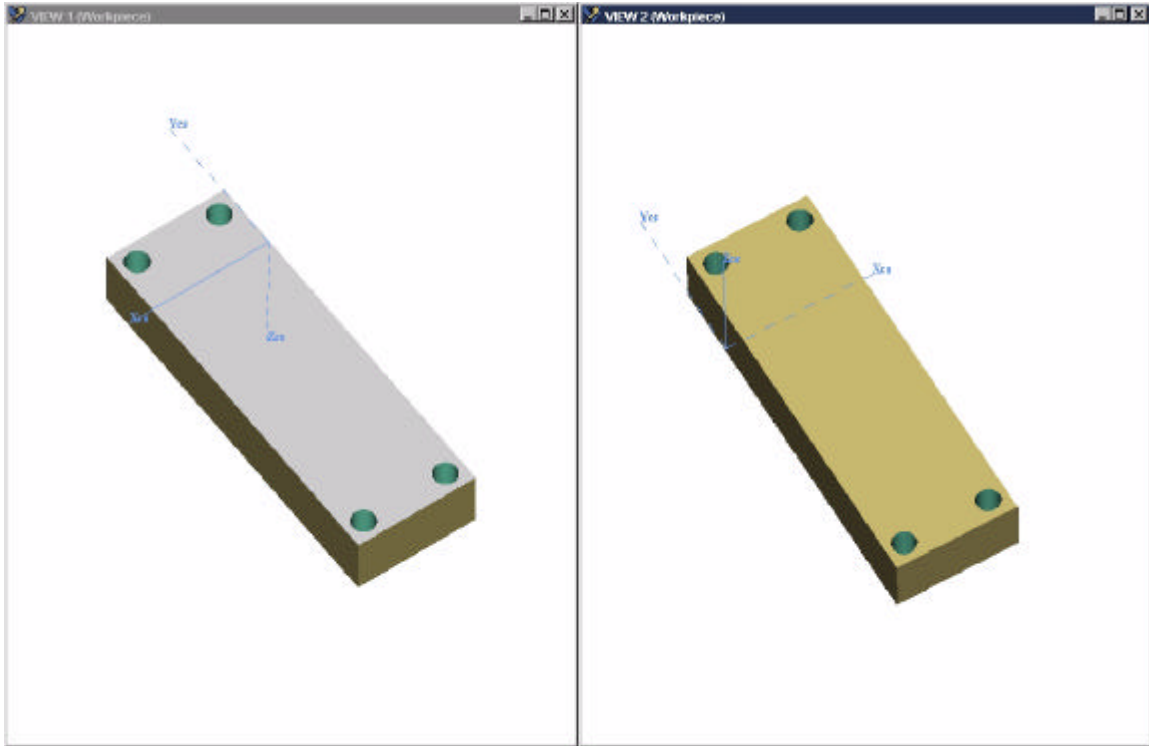
Click on **Position** (when the field is highlighted, the position can be picked on the model or typed in the field)

Select the top left corner



Apply

Translate From=0 0 0, To=0 246 50, Move



10. Add "flange2.tp" and associate the coordinate systems to each toolpath

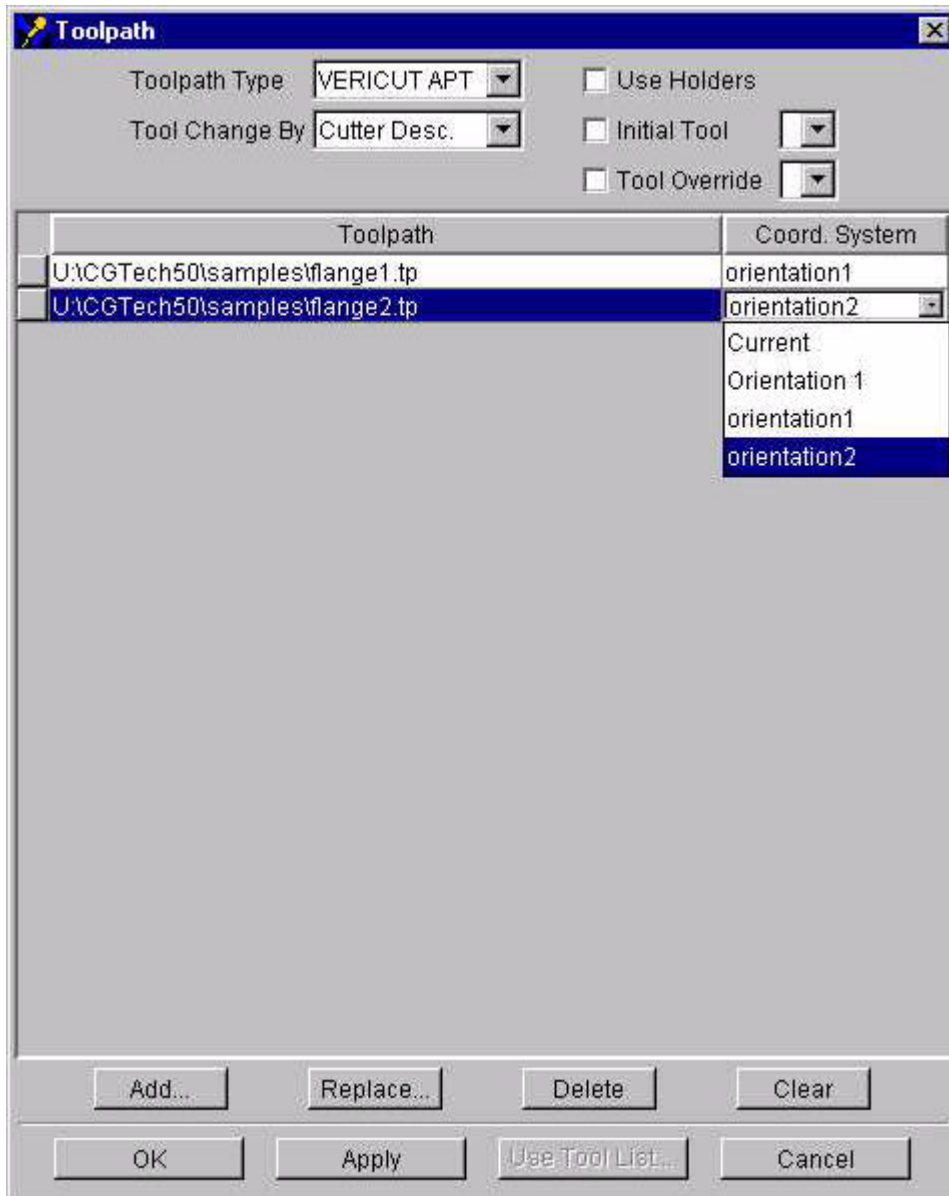
Setup > Toolpath

Add

Shortcut=CGTECH_SAMPLES

File Name=flange2.tp

Click in the field under **Coordinate System**, **Coordinate System=orientation 2**



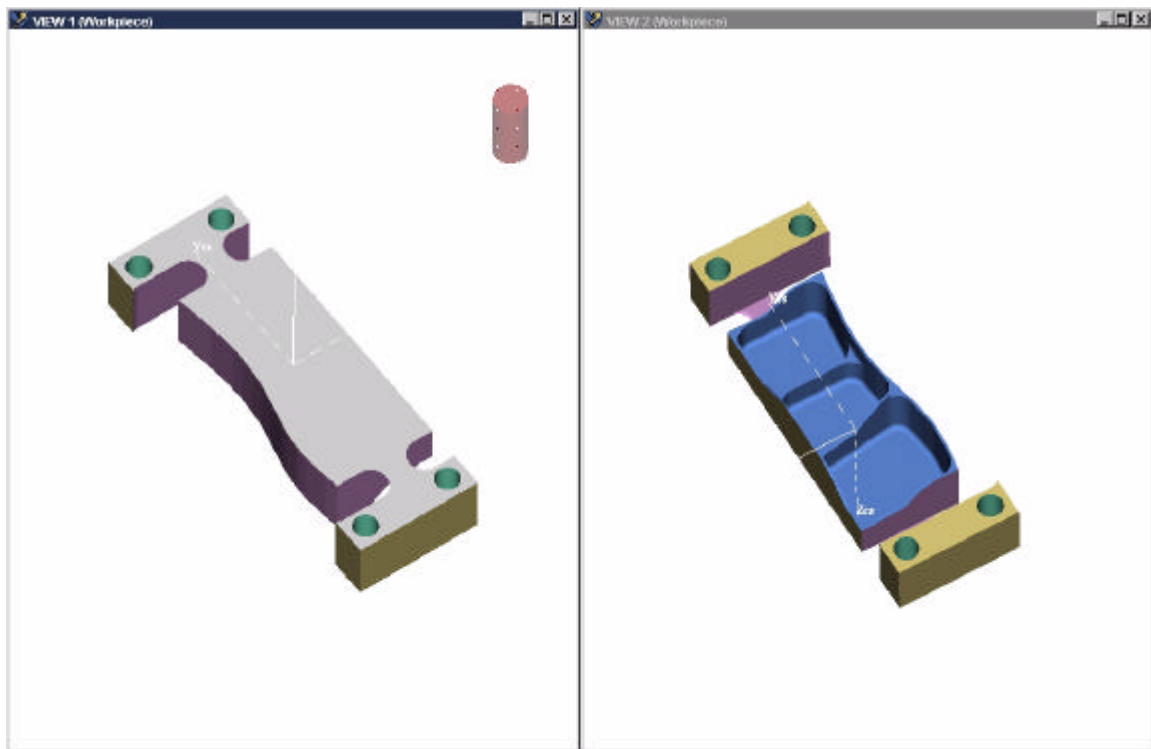
Click on **flange1.tp**, Coordinate System=**orientation 1**

OK

11. Cut the part in both positions

Reset Model 

Play to End 



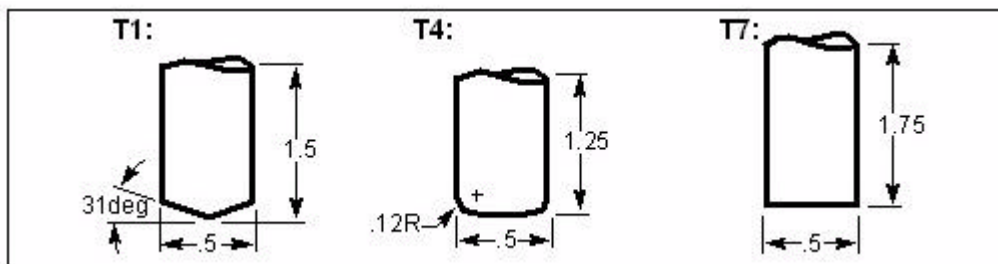
12. Exit the VERICUT session

File > Exit, No

Session 7- Create a Tool Library

This session shows how to use the **Tool Manager** function to graphically define cutter shapes and store them in a Tool Library file.

Assume a G-code tool path file to be simulated requires the cutter shapes shown below.



Session Steps:

1. Start with a new Inch User file

File > Properties

Default units=Inch, OK

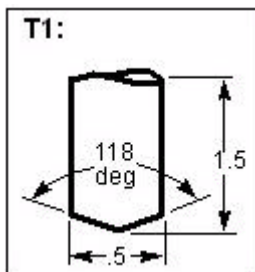
File > New

2. Access the Tool Manager and specify that a new Tool Library file is being created

Setup > Tool Manager

File > New

3. Add a .5 dia., 1.5 ht., 118 deg. drill as tool library ID "1" (used by "T1")



Add

ID=1

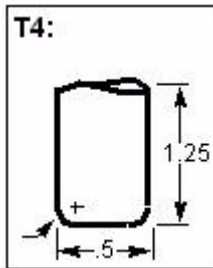
Description=.5D 1.5H DRL



Diameter (D)=.5, **Drill Point Angle (A)**=118, **Height (H)**=1.5, **Add** (the drill is displayed in the Tool Manager window)

OK

4. Add a .5 dia., .12 cr, 1.25 ht. end mill as ID "4" (used by "T4")



Add

ID=4

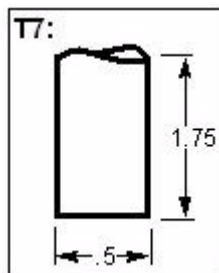
Description=.5D .12R 1.25H EM



Diameter (D)=.5, **Corner Radius (R)**=.12, **Height (H)**=1.25, **Add**

OK

5. Add a .5 dia., 1.75 ht. flat end mill as ID "7" (used by T7)



Add

ID=7

Description=.5D 1.75H FEM



Help on VERICUT

Diameter (D)=.5, Height (H)=1.75, Add

OK

6. Save the tools in a Tool Library file named "new.tls" and close the Tool Manager

File > Save As

Shortcut=Working Directory

File=new.tls, Save

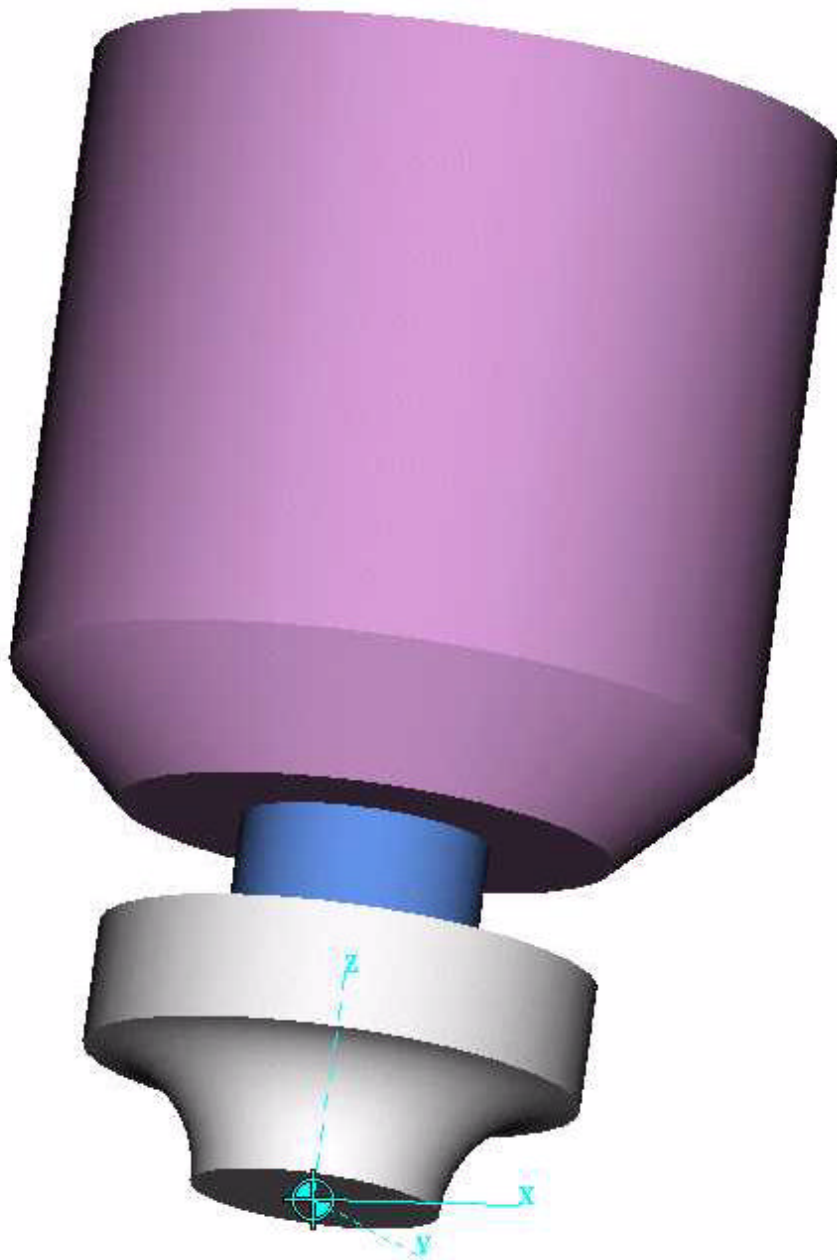
File > Close

VERICUT can later be configured to process a G-code tool path file and use this library as the source of tool descriptions.

Session 8- Mill tool profiles and a default tool

This session shows how to use the **Tool Manager** function to define a tool assembly (cutter and holder) via profile descriptions. The tool is specified for use in VERICUT via the **Override** feature as a default tool. Supplying a tool description by tool override is required when the tool path file does not contain tool descriptions, or can be useful to try a temporary "test tool".

This session demonstrates defining a "corner rounding" mill tool assembly similar to that shown below. The **Tool Manager** allows tools having almost any shape to be easily defined, as well as provides collision detection between non-cutting portions of the tool assembly and the workpiece.



Session Steps:

1. In VERICUT, open the "vcblock.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=vcblock.usr, OK, OK (If a message appears, press **Reset.**)

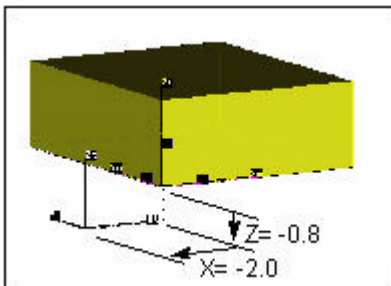
2. To improve visibility of tool profiles, rotate the view another 50 degrees X-

View > Orient

Increment=50, X-



3. Orient the Coordinate System as shown below



Setup > Coordinate System

Coordinate System Name=orientation 1

Position= -2 0 -.8, Apply

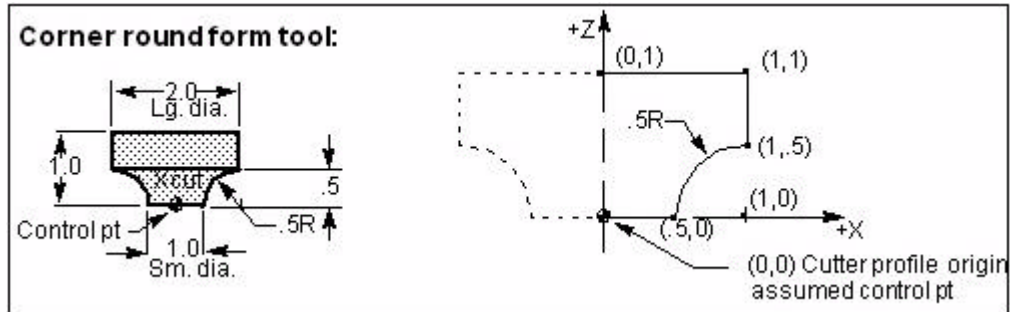
OK

4. Access Tool Manager and create a new Tool Library file

Setup > Tool Manager

File > New

5. Use the Profile option to describe the corner round form tool shown below (use tool ID "999")



Add

ID=999

Description=CORNER RND EM



Point, Append - adds X= 0 , Z= 0

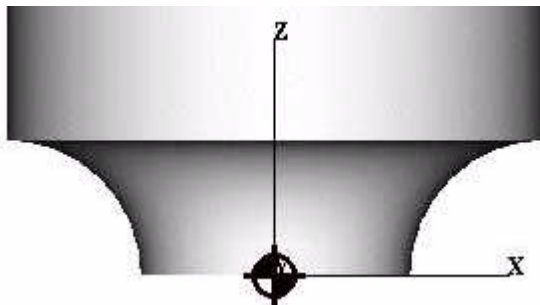
Point, X=.5, Z=0, Append

Arc, X Center=1, Z Center=0, Radius=.5, Append

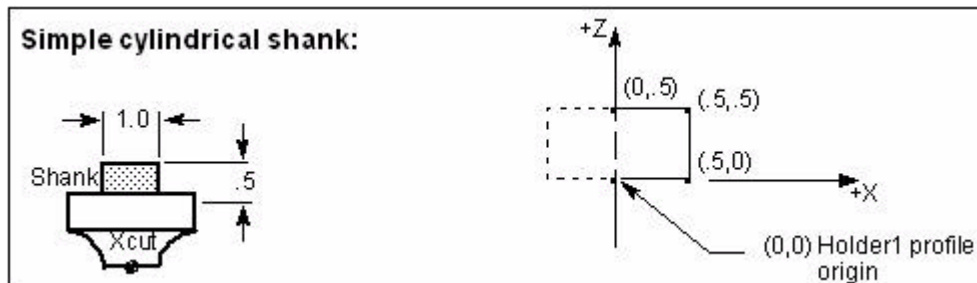
Point, X=1, Z=.5, Append

Point, X=1, Z=1, Append

Add



6. Describe the 1.0 diameter shank of tool ID "999"



Tool Component=Holder1

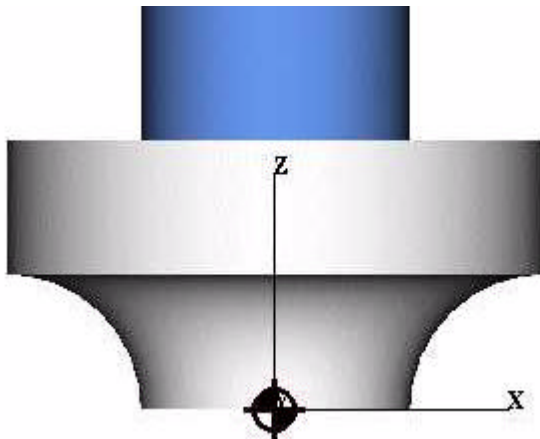


Point, Append - adds $X = 0, Z = 0$

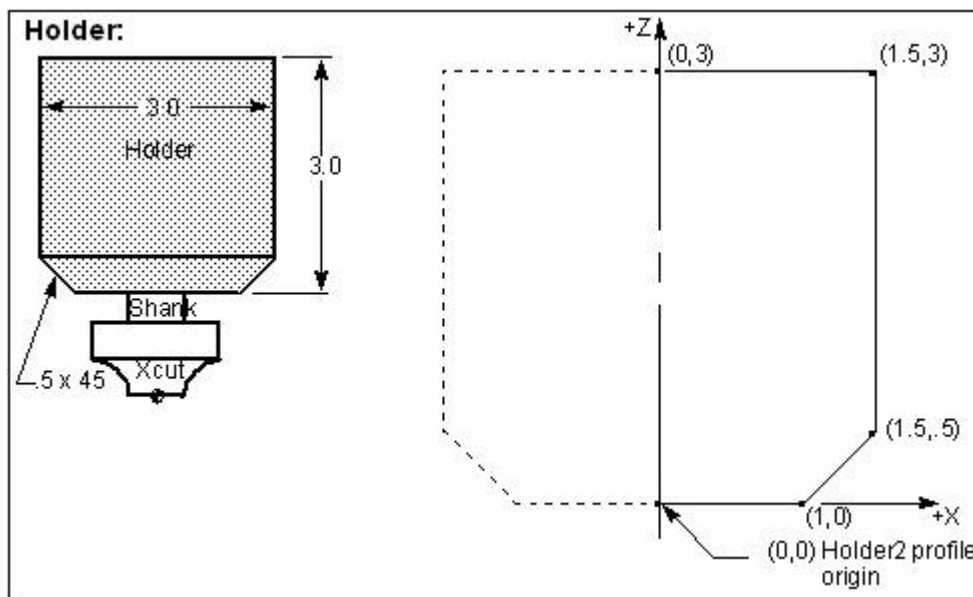
Point, $X=.5, Z=0$, Append

Point, $X=.5, Z=.5$, Append

Add



7. Describe the 3.0 diameter holder of tool ID "999", then add the tool to the library



Tool Component=Holder2



Profile

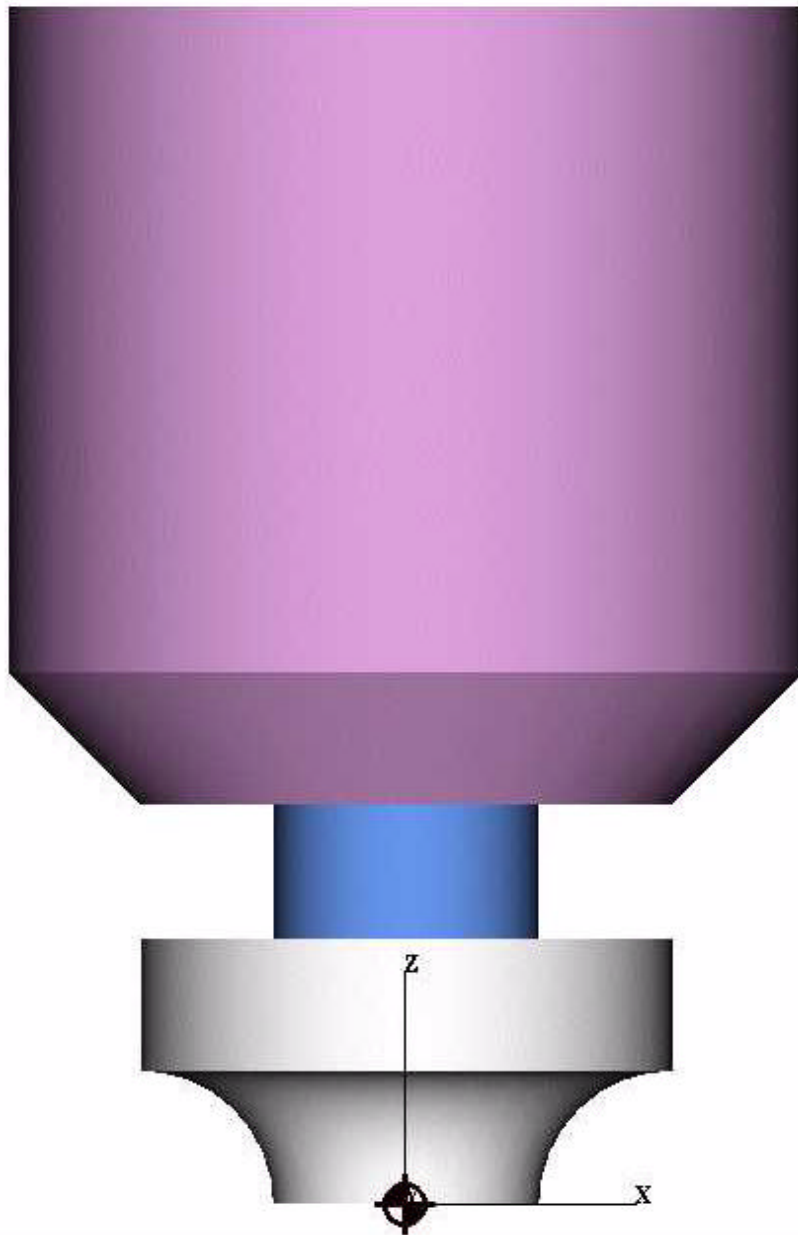
Point, Append - adds $X=0$, $Z=0$

Point, $X=1.0$, $Z=0$, Append

Point, $X=1.5$, $Z=.5$, Append

Point, $X=1.5$, $Z=3$, Append

Add, OK



8. Save the tool in a Tool Library file named "special.tls" and close the Tool Manager

File > Save As

Shortcut=Working Directory

File Name=special.tls, Save

File > Close, Yes

9. Override tool changes to make Tool Library tool "999" the active tool

Setup > Toolpath

Tool Override=999, OK

10. Cut the model- how many errors were detected?

Play to End 



Select **Info > Status**, Error=0, but the cutter went so deep that it was submerged under material

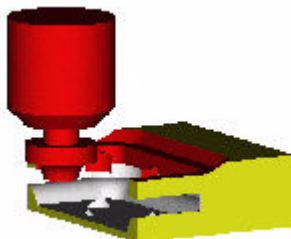
13. Activate the holder override for tool "999", then reset and cut the model

Setup > Toolpath

Select **Use Holders, Apply**

Reset Model 

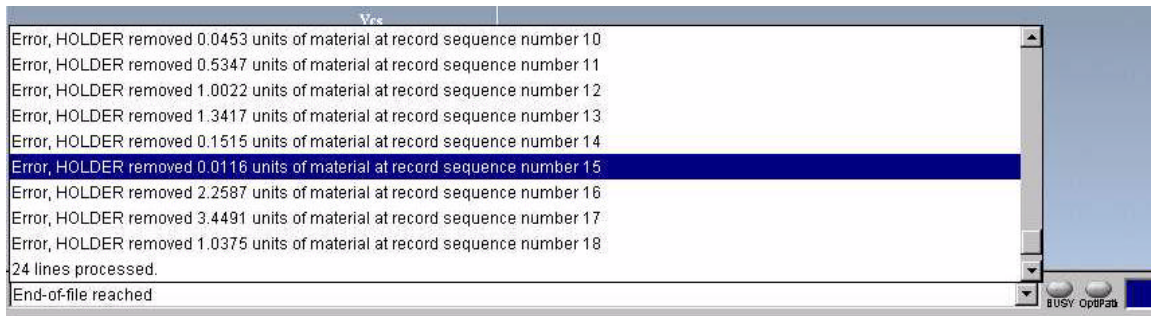
Play to End 



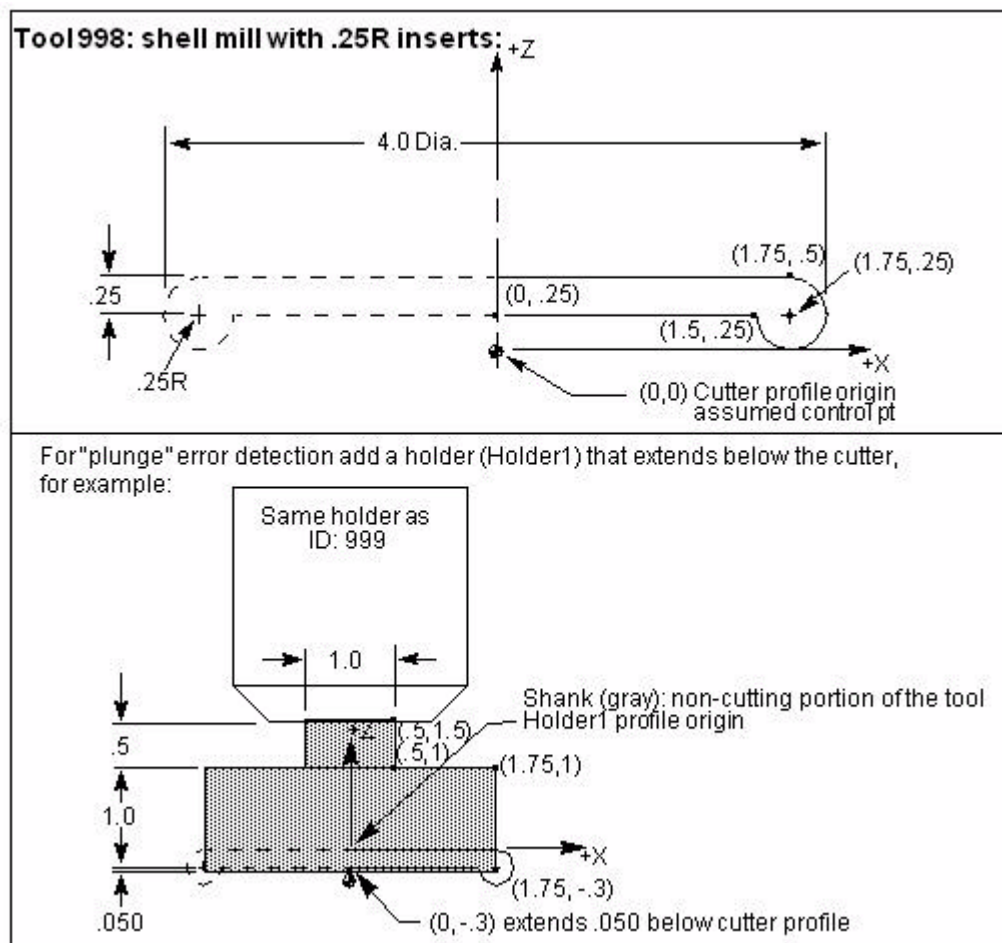
14. How many and what type(s) of errors were detected?

Select **Info > Status**, Error=9

Scroll up the **Message Area** to see the error messages



15. Add tool "998" to the Tool Library to represent the carbide insert shell mill shown below (use the holder from ID "999")



Cutter:

Help on VERICUT

Setup > Tool Manager

Add

ID=998

Description=4.0D .25R SHELL MILL ,

Profile



Point, X=0, Z=.25, Append

Point, X=1.5, Z=.25, Append

Arc, X Center=1.75, Z Center=.25, Radius=.25, Direction=CCW, Append

Point, X=1.75, Z=.5, Append

Point, X=0, Z=.5, Append

Add



Shank:

Tool Component=Holder1



Profile

Point, X=0, Z=-.3, Append

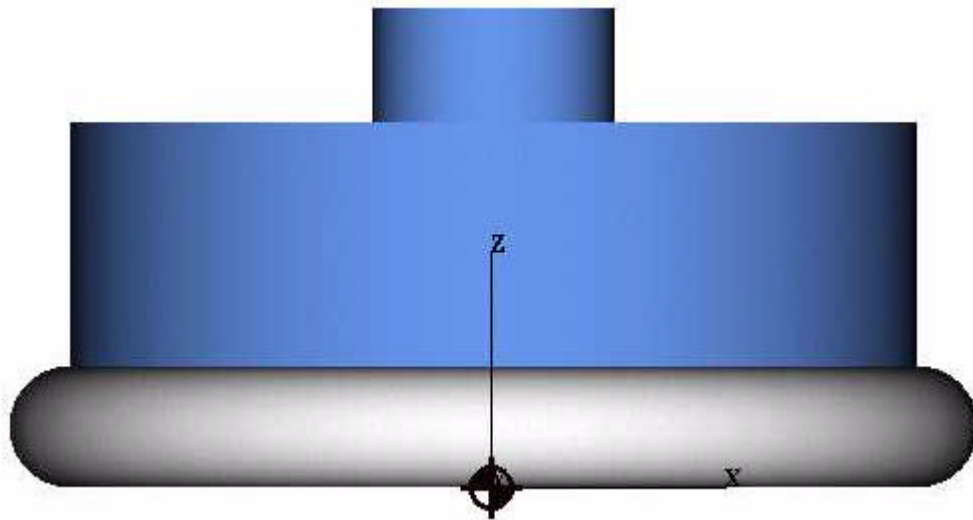
Point, X=1.75, Z=-.3, Append

Point, X=1.75, Z=1, Append

Point, X=.5, Z=1, Append

Point, X=.5, Z=1.5, Append

Add



Reference the holder in tool ID 999:

Tool Component=Holder2



Reference

Reference ID=999

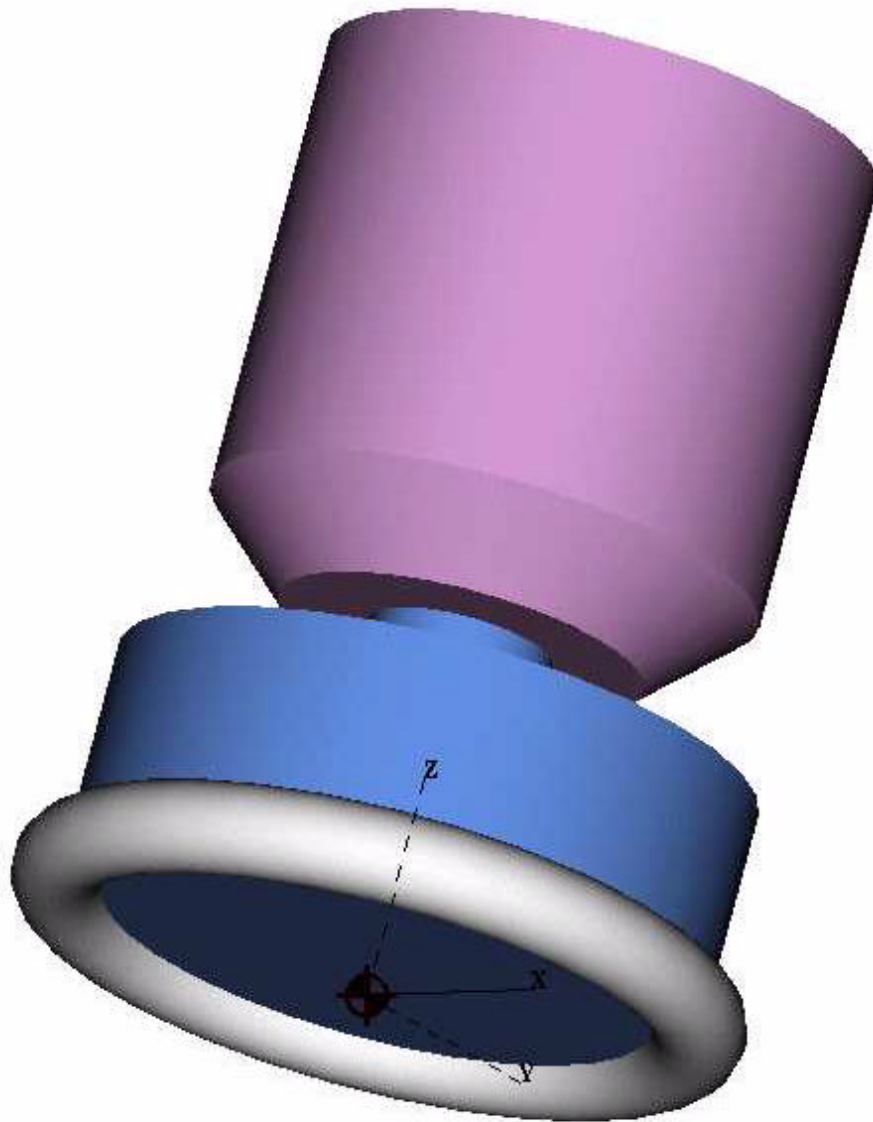
Add,

View > Orient

Angles= -115 40 5, Apply

Close

In the view below you can see the non-cutting collision zone under the tool. An error is caused if that portion of the tool assembly removes material.



16. Save the Tool Library, set tool Override to tool "998" , then cut the model

File > Save

File > Close

Setup > Toolpath

Tool Override=998, OK

Reset Model 

Help on VERICUT

Play to end



Note the plunge error "Error, Holder removed material at record sequence number 18" issued when the shell mill ramps into material.

Session 9- Use a tool list and "master" Tool Library

This session shows how to use the **Tool Manager** function to define cutter shapes and store them in a "master" Tool Library file. Master tool libraries typically have tool identification (ID) values that differ from the "T" numbers in the G-code tool path file. A tool list is used to cross-reference G-code tool change blocks to access tool descriptions stored in the Tool Library file. This feature makes it possible to define a single Tool Library file containing all available tool descriptions, and have all users use this file as a source of tool descriptions for VERICUT.

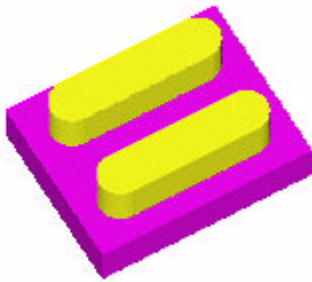
Session Steps:

1. In VERICUT, open the "bars.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=bars.usr, OK, OK (If a message appears, press **Reset.**)



File > Properties

Default units=Millimeter, OK

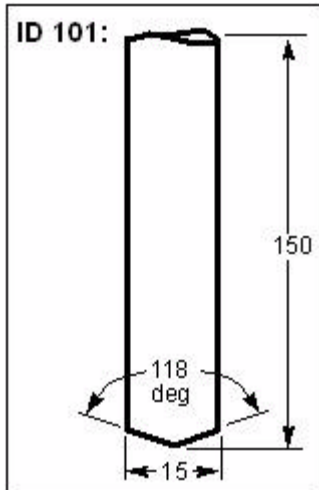
This assures that the units for a new Tool Library are set to millimeter.

2. Access the Tool Manager and create a new Tool Library file

Setup > Tool Manager

File > New

3. Add tool ID "101": 15 dia., 150 ht., 118 deg. drill



Add

ID=101

Description=15D 150H DRL

Ensure **Units=Millimeter** (if they are **Inch**, then redo the previous step)



Drill

Diameter (D)=15

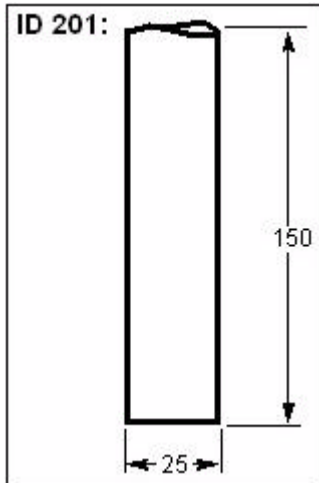
Drill Point Angle (A)=118

Height (H)=150

Add (the drill is displayed)

OK

4. Add tool ID "201": 25 dia., 150 height, flat bottom endmill



Add

ID=201

Description=25D 150H FEM

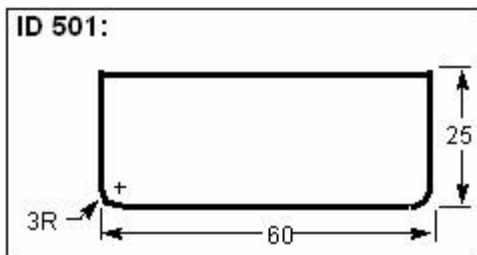


Flat End Mill

Diameter (D)=25, Height (H)=150, Add

OK

5. Add tool ID "501": 60 dia., 3 cr., 25 ht. end mill



Add

ID=501

Description=60D 3R 25H EM



Bull Nose End Mill

Diameter (D)=60, Corner Radius (R)=3, Height (H)=25, Add

OK

6. Save the tools in a Tool Library file named "master.tls" and close the Tool Manager

File > Save As

Shortcut=Working Directory

File Name=master.tls, Save

File > Close, Yes

7. Build a tool list that cross-references G-code tool numbers to tools in the Tool Library as follows: T1M6 uses ID101, T2M6 uses ID201, T3M6 uses ID501

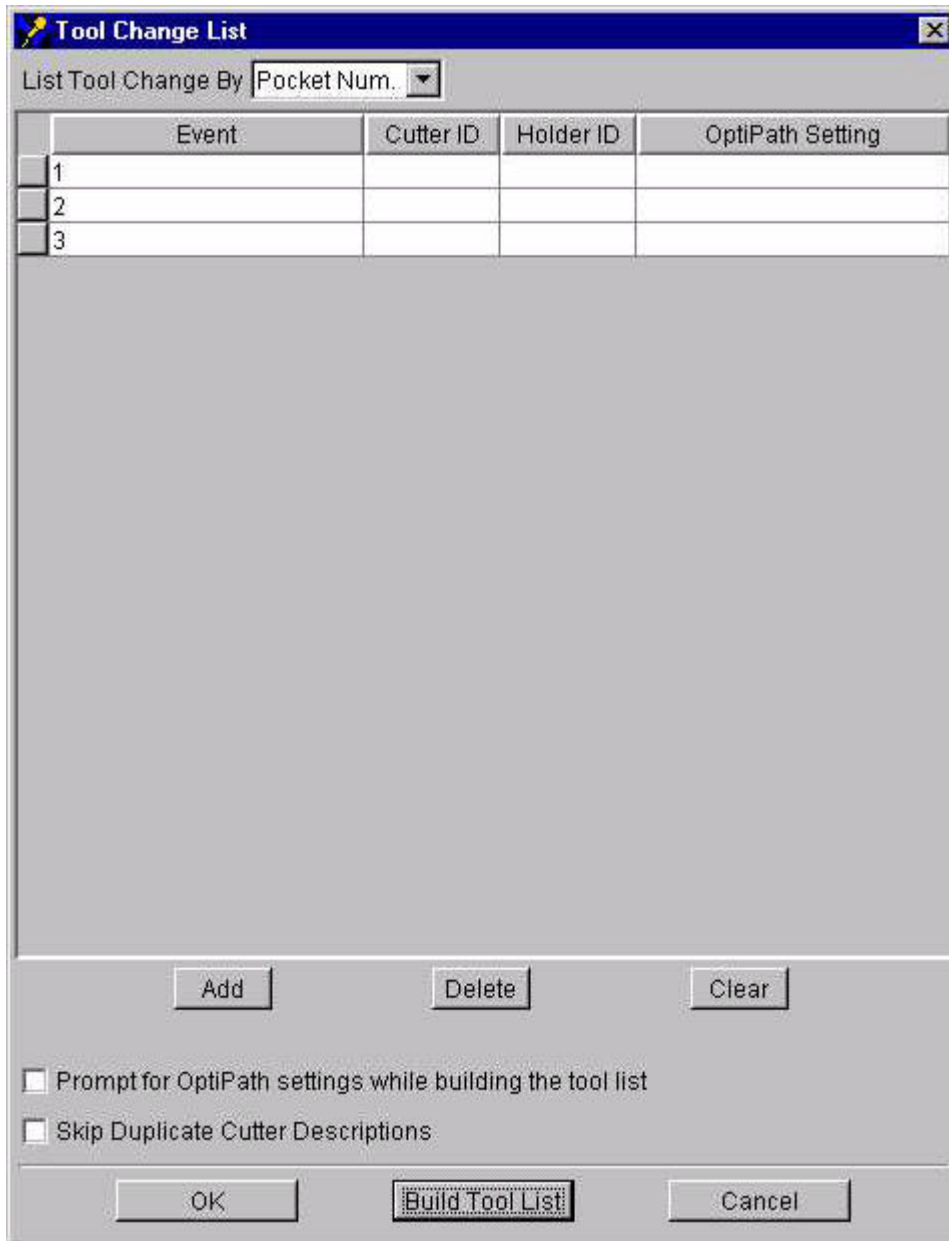
Setup > Toolpath

Tool Change By=List

Use Tool list

Make sure **Prompt for Optipath** is NOT selected

Build Tool List -> scans the tool path and generates the following tool list based on tool pocket numbers:

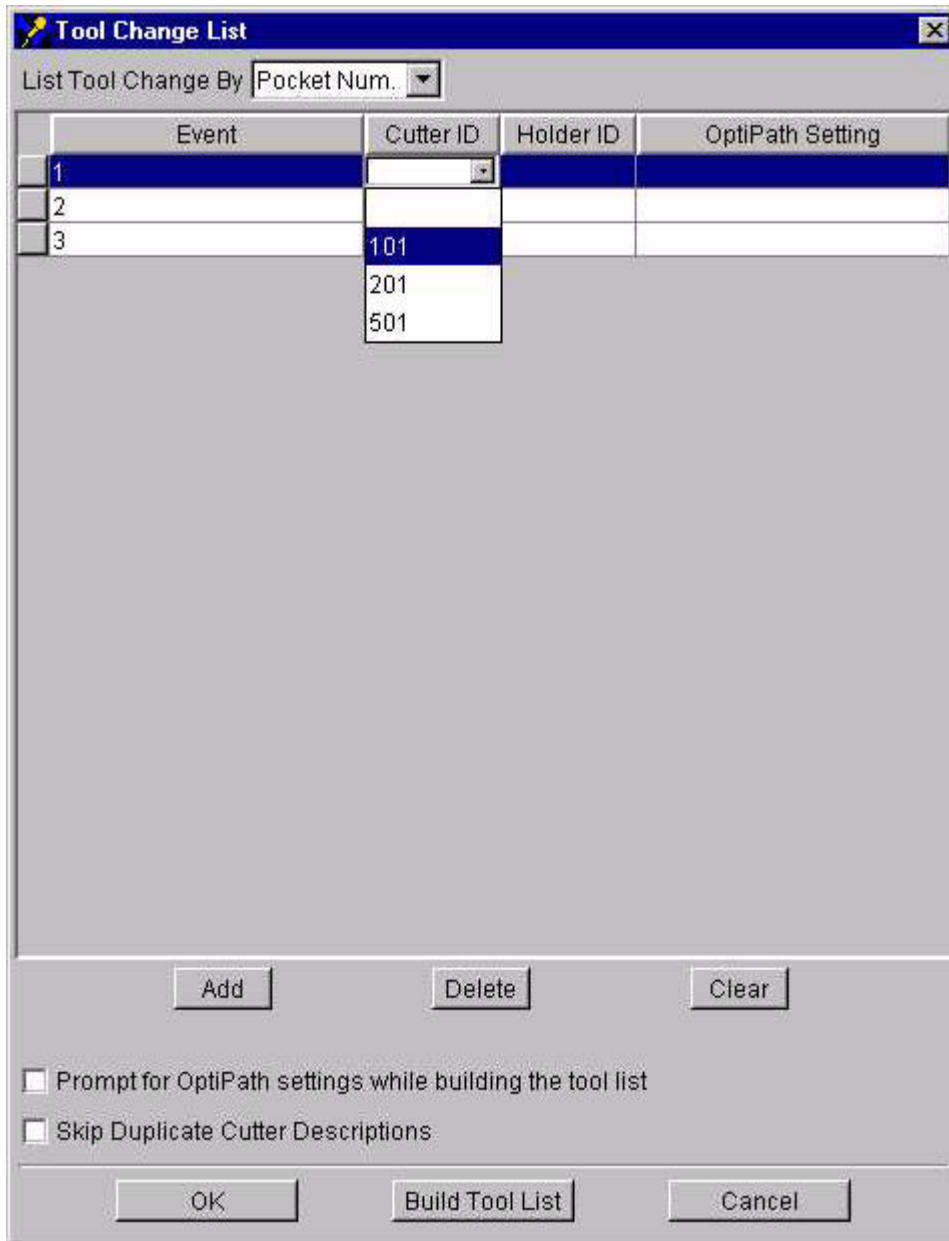


The image shows a 'Tool Change List' dialog box. At the top, there is a title bar with a key icon and the text 'Tool Change List'. Below the title bar, there is a label 'List Tool Change By' followed by a dropdown menu showing 'Pocket Num.'. The main area of the dialog contains a table with four columns: 'Event', 'Cutter ID', 'Holder ID', and 'OptiPath Setting'. The table has three rows, numbered 1, 2, and 3 in the first column. Below the table, there are three buttons: 'Add', 'Delete', and 'Clear'. At the bottom, there are two checkboxes: 'Prompt for OptiPath settings while building the tool list' and 'Skip Duplicate Cutter Descriptions'. At the very bottom, there are three buttons: 'OK', 'Build Tool List', and 'Cancel'.

	Event	Cutter ID	Holder ID	OptiPath Setting
1				
2				
3				

☐ Prompt for OptiPath settings while building the tool list
☐ Skip Duplicate Cutter Descriptions

With **Event 1** (T1M6) selected, enter **Cutter ID=101**



Select **Event 2 (T2M6)**, **Cutter ID=201**

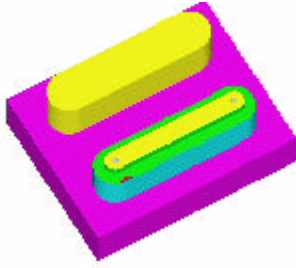
Select **Event 3 (T3M6)**, **Cutter ID=501**

OK, OK

8. Cut the model

Reset Model 

Play to End 



Session 10- Visually analyze a cut model

This session shows how to configure VERICUT for re-processing a tool path file via opening a previously saved User file, then cutting. A User file typically has all of the information required to configure VERICUT for processing a tool path file: stock description, tool path file name and orientation, where to look for cutting tool descriptions, etc. During this session the cut model will be visually analyzed using functions that rotate, section, and magnify the model. Finally, the model will be saved in an In Process file (IP file) for later use.

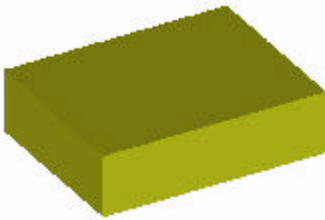
Session Steps:

1. Open the "vcmill.usr" User file

File > Open

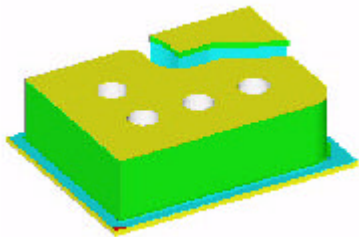
Shortcut=CGTECH_SAMPLES

File Name=vcmill.usr, OK, OK (If a message appears, press **Reset.**)



2. Cut the VERICUT stock model

Play to End 



The **View Orient** functions manipulate the model so it can be seen from various distances or angles. These actions can be extremely helpful when analyzing a model.

The **Reverse** viewing feature enables the model to be seen from the opposite side. This gives the same appearance as if the user walked to the rear of the computer screen and viewed the model from behind.

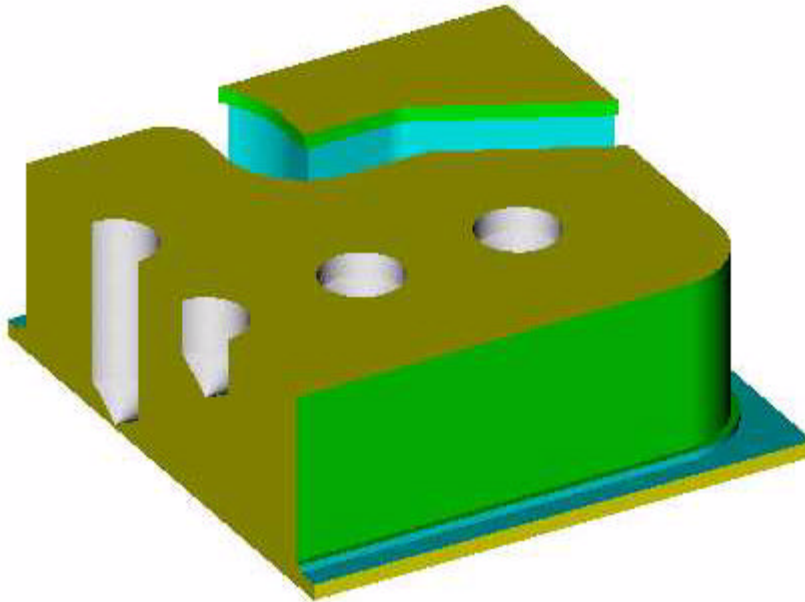
3. Section (cross-section) the model through the 2 in-line holes

View > Section

Select **Section on mouse pick**

Ensure Plane 1 is set to **Plane Type=X**

Left-click on the model between the 2 holes drilled on the left side (see below)



4. Zoom (magnify) a portion of the model, then restore from the zoomed view

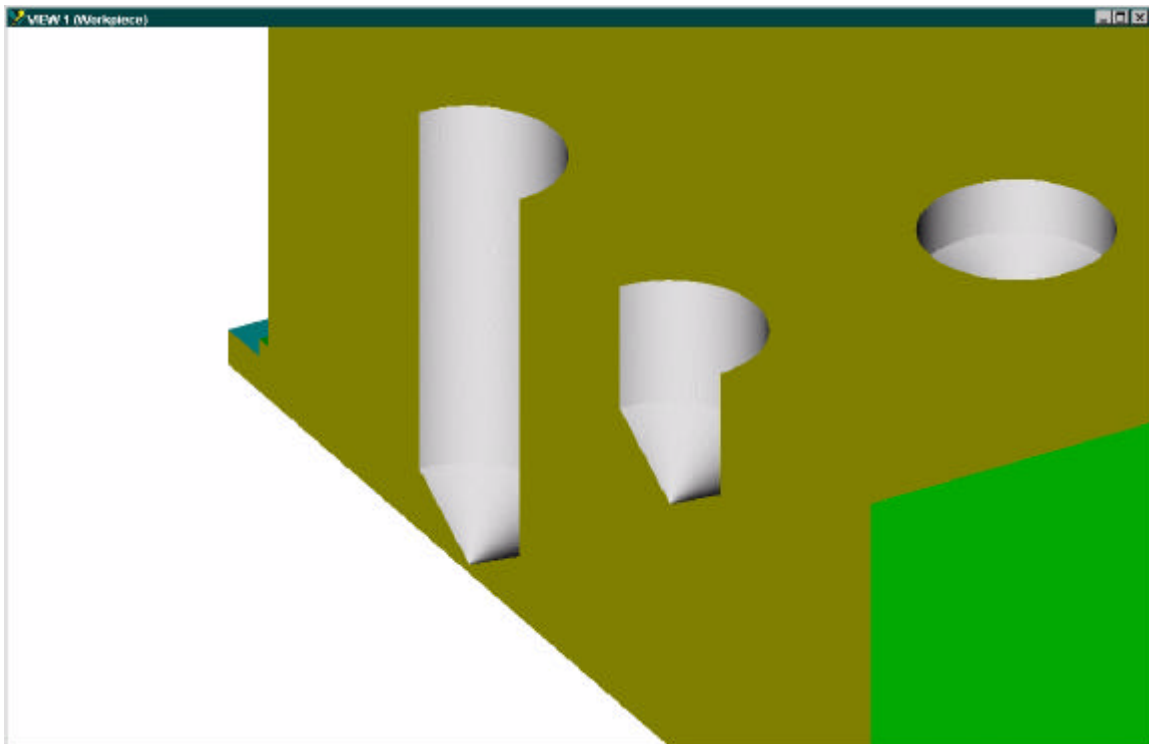
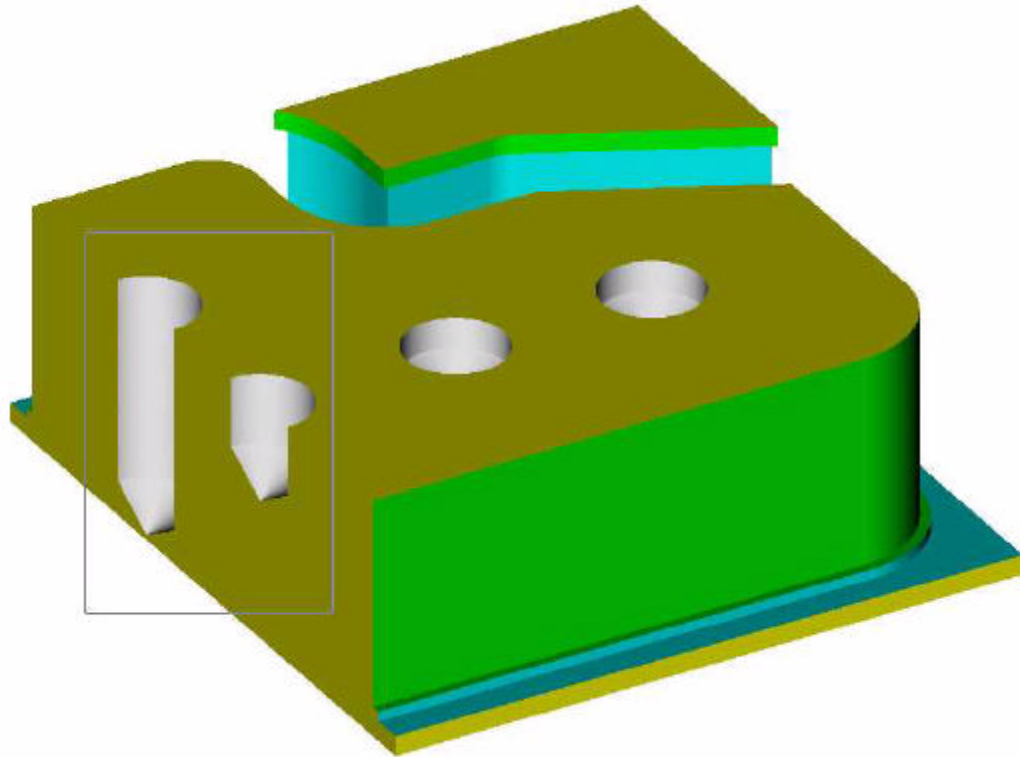
The **Zoom** viewing feature magnifies a portion of the model. A rubberbanding point box indicates the target area to zoom.

View > Orient

Zoom - Left-click at the location to anchor the rubberband box (see below left)

Stretch the rubberband box around the target area to be zoomed, then click again to start the zoom

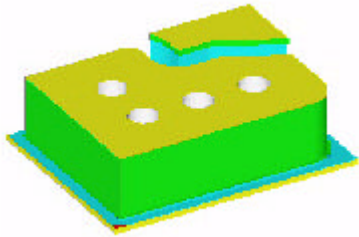
Fit



5. Restore the model from the sectioned view

View > Section

Restore



Saving an IP file enables you to re-load the VERICUT model for additional analysis or machining. An IP file saves all information required to restore the current VERICUT session is saved, including: the current cut model, configuration settings (known as "user values"), and the current VERICUT Log File with any errors, warnings or other informational messages logged during the VERICUT session.

6. Save the VERICUT model in an In-process file named "session3.ip"

File > In Process > Save As

Shortcut=Working directory

File name=session10.ip, Save

IP files are primarily used to save the current VERICUT model, for example saving a "rough machine" model prior to simulating finish machining motions. However, IP files are also useful to "backup" the VERICUT session safeguarding against losing work when the session is interrupted, such as occurs during a computer system or power failure.

7. Exit VERICUT

File > Exit, Yes

This session demonstrated how to configure VERICUT for re-processing a tool path file via loading a previously saved User file. After displaying and cutting the stock model, **View** functions were used to rotate, section, and zoom on the cut model. The step prior to exiting showed how to save the current VERICUT session (including the cut model, session configuration, etc.) via saving an IP file. The session is restored by loading the IP file. After loading the IP file, the VERICUT session appears exactly as it did when saved.

Session 11- Measure a VERICUT Model Using X-Caliper

This session requires using the X-Caliper function to measure and inspect a cut model.

Session Steps:

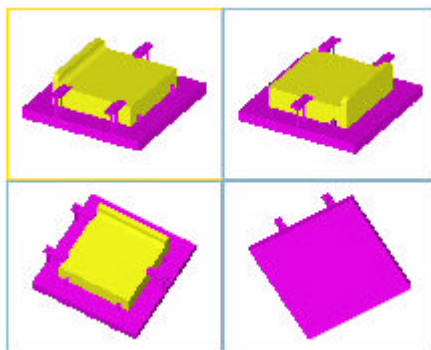
1. Open the "vc5side.usr" User file

File menu > Open

Shortcut=CGTECH_SAMPLES

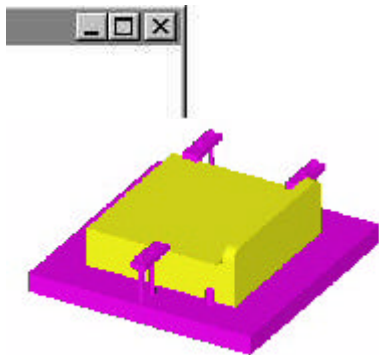
File Name=vc5side.usr > OK > OK

If a message appears, press **Reset**.



2. Close View 1,3 & 4

Close views using the X icon on the top right of each view.

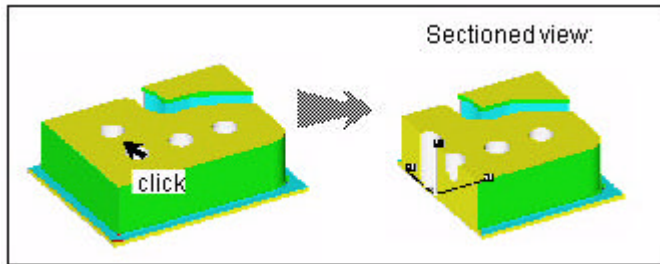


3. Cut the model - how many and what kinds of errors did VERICUT detect?

Play to end 

Determine the quantity of errors detected: **Info > Status** - 15 errors

Determine the kinds of errors detected: **Info > VERICUT Log** -> see the Log file contents



Dimensions to Measure on a Sample Cut Model

4. Measure the cut model to determine values for dimensions "a" - "m" above

Analyse > X-Caliper

To save space, only the correct dimensions are shown below with the **X-Caliper Tab** used to extract them from the cut model. *Values are rounded to the nearest millimeter.*

a=20 **Stock Thickness Tab**

b=20 **Stock Thickness Tab**

c=360 **Air Distance Tab**

d=20 **Stock Thickness Tab**

e=150 **Air Distance Tab**

f=360 **Air Distance Tab**

g=20 **Stock Thickness Tab**

h=60 **Stock Thickness Tab**

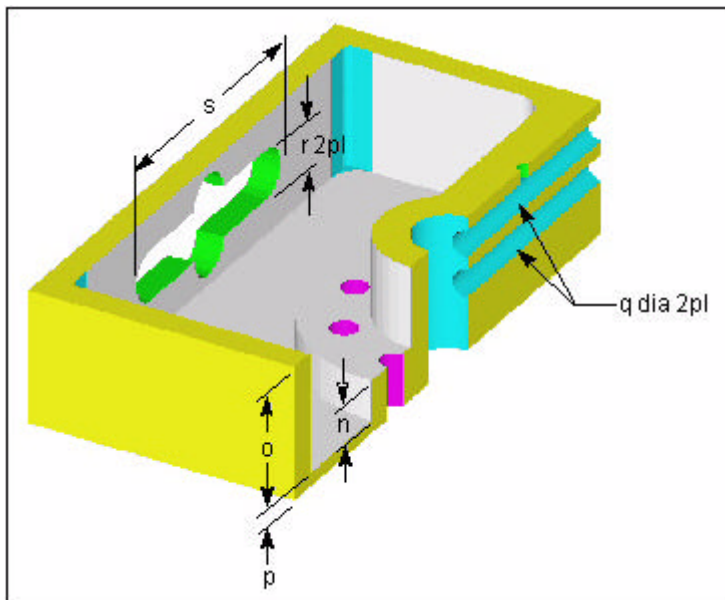
i=50 **Air Distance Tab** or **Feature/History Tab**

j=100 **Air Distance Tab** + 2x **Stock Thickness Tab**

k=75 **R Distance/Angle Tab: Axis-Point**

l=25 **R Feature/History Tab**

m=20 **Air Distance Tab** or **Feature/History Tab**



More Dimensions to Measure on a Sample Cut Model

5. Measure to determine values for "n" - "s" above

n=30 **Distance/Angle Tab: Plane-Plane**

o=90 **Distance/Angle Tab: Plane-Plane**

p=10 **Stock Thickness Tab**

q=20 **Air Distance Tab or Feature/History Tab**

r=37.5 **Air Distance Tab**

s=187 **Air Distance Tab**

Session 12- AUTO-DIFF and Comparing a Design Model

This session shows how to use the AUTO-DIFF function to compare a design model against the VERICUT simulated cut part and detect gouge or excess material conditions. Solid, surface, or inspection point design model data can be used by AUTO-DIFF to clearly identify where discrepancies between the VERICUT model and the design model exist.

AUTO-DIFF identifies these discrepancies:

gouge conditions are shaded using the **Error** color (typically red)

excess material conditions are shaded using the **Fixture** color (typically magenta)

*Hint: Use other VERICUT functions, such as **X-Caliper** and **Zoom**, to help determine the source of discrepancies displayed by an AUTO-DIFF operation.*

Session Steps:

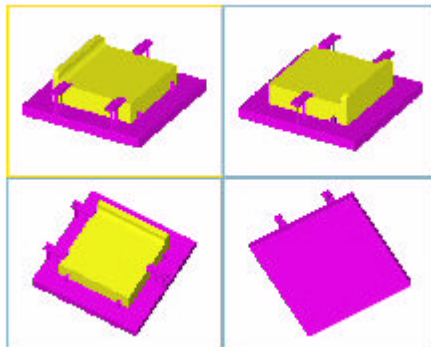
1. In VERICUT, open the "vc5side.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=vc5side.usr > OK > OK

If a message appears, press **Reset**.

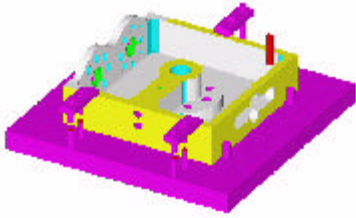


2. Display the VERICUT model in a single view layout and cut -> how many and what kinds of errors does VERICUT detect?

View > Layout > Standard

click on **1 views**

Play to end 



Determine the quantity of errors detected: **Info > Status** ->15 errors

Determine the kinds of errors detected: **Info > VERICUT Log** -> see the Log file contents

3. Remove the displayed fixture model

Model > Model Definition:Model tab

Click on fixture model

Model Definition:Component Attributes tab

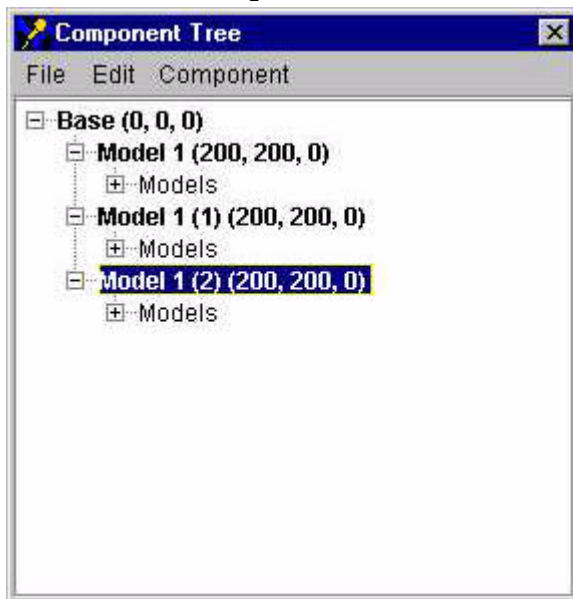
Visibility=Blank > OK > Yes

4. Find the "vc5side.dsn" VERICUT Polygon model in the ComponentTree

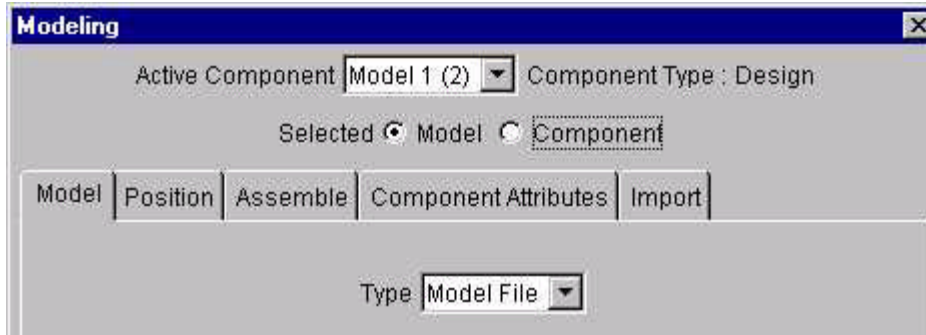
Model > Component Tree

Model > Model Definition:Model tab

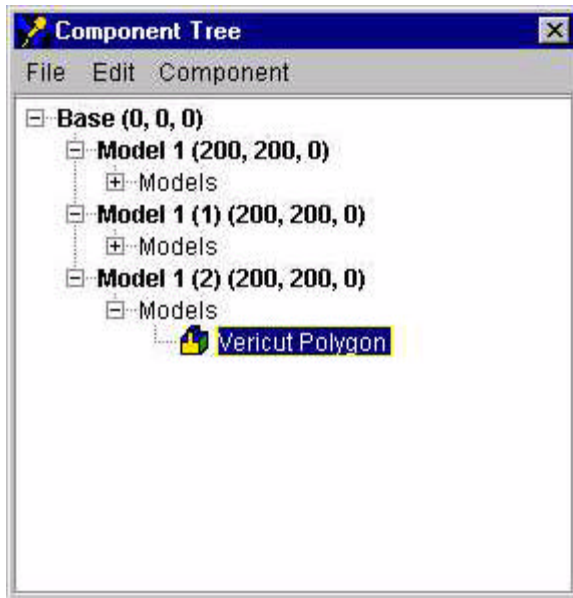
In the **Component Tree** click on **Model 1(2)**



In the **Modeling** window **Selected=Model**



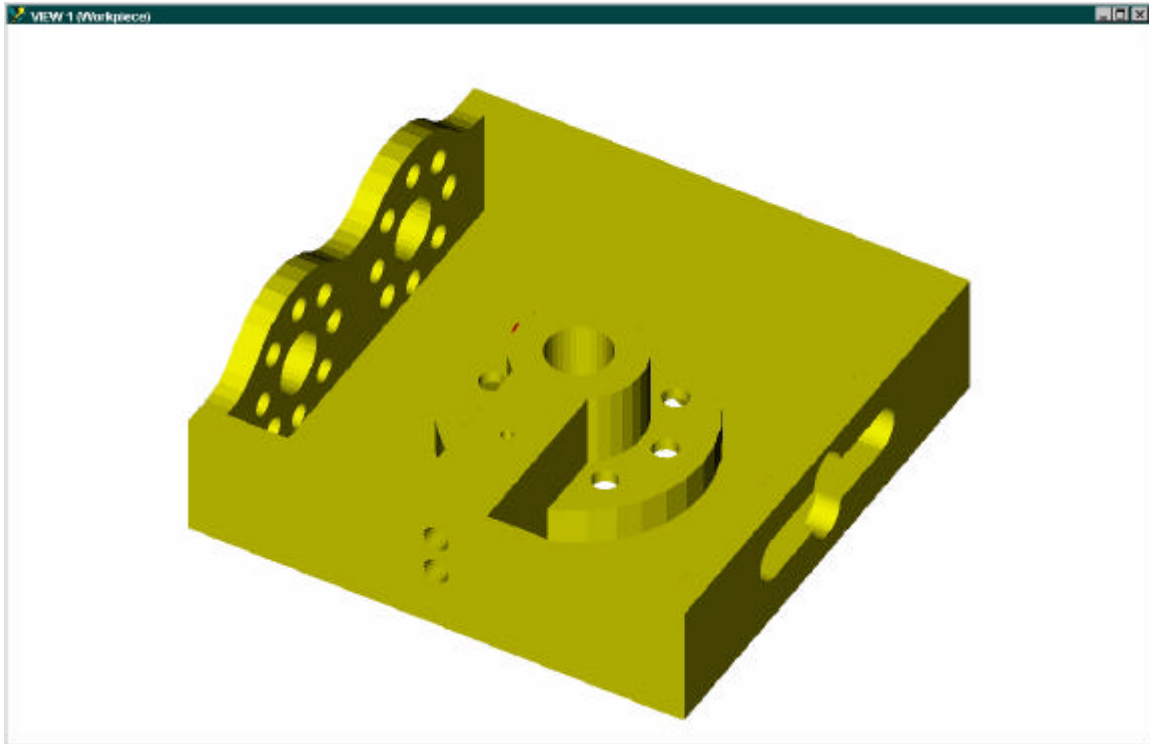
(notice the **Vericut Polygon** is selected in the **ComponentTree** window)



In the **Modeling** window, **Close**

**5. Compare the design model to check for gouges deeper than .25 mm;
use an option that shows the design model with any gouges in it**

Analysis > AUTO-DIFF
Tolerance Tab
Gouge=.25
Excess=.25
Apply
Setting Tab
Comparison Type=Gouge
Click on Design Display
Compare



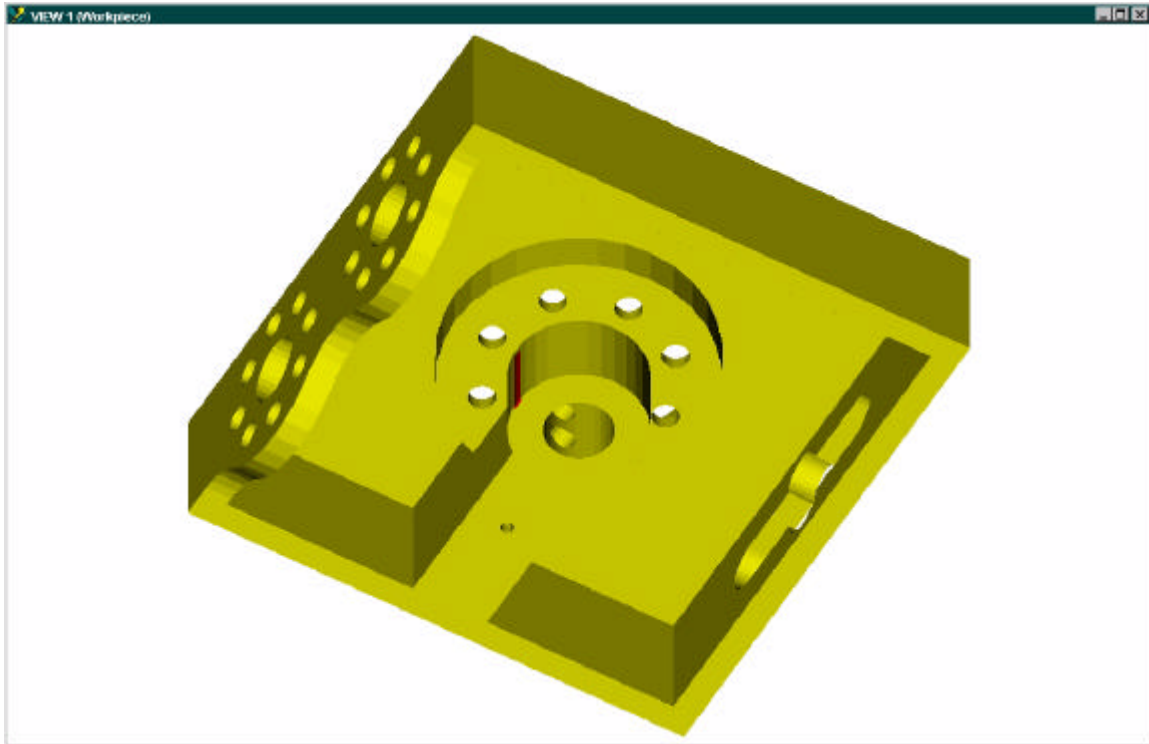
6. Rotate the VERICUT model to see if gouges are present-> are there?

At first glance, the resulting **AUTO-DIFF** model does not appear to have any gouges. However, looking at the opposite side of the model reveals that there is a gouge!

Dynamic X Rotation > Drag the cursor downward to rotate the part as shown

below 

Refine Display 



Notice the red mark

7. What tool path record caused the gouge?

Click on **Design Display** (only a thin slice of material is visible)

Analysis > X-Caliper:Feature/History Tab

Reverse 

Pick on the red gouge material

Close

8. Restore from the AUTO-DIFF operation, then compare the design model to check for excess material beyond .25 mm- is there any?

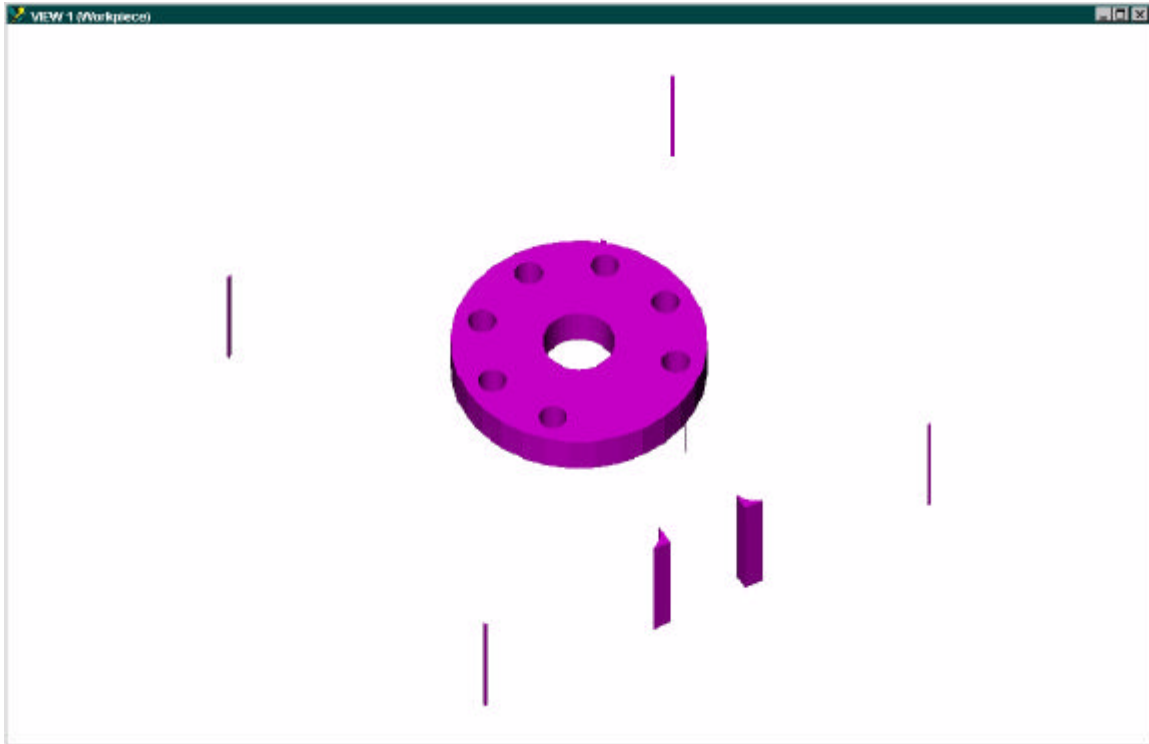
Analysis > AUTO-DIFF

Restore

Comparison Type=Excess

Compare

Refine Display 



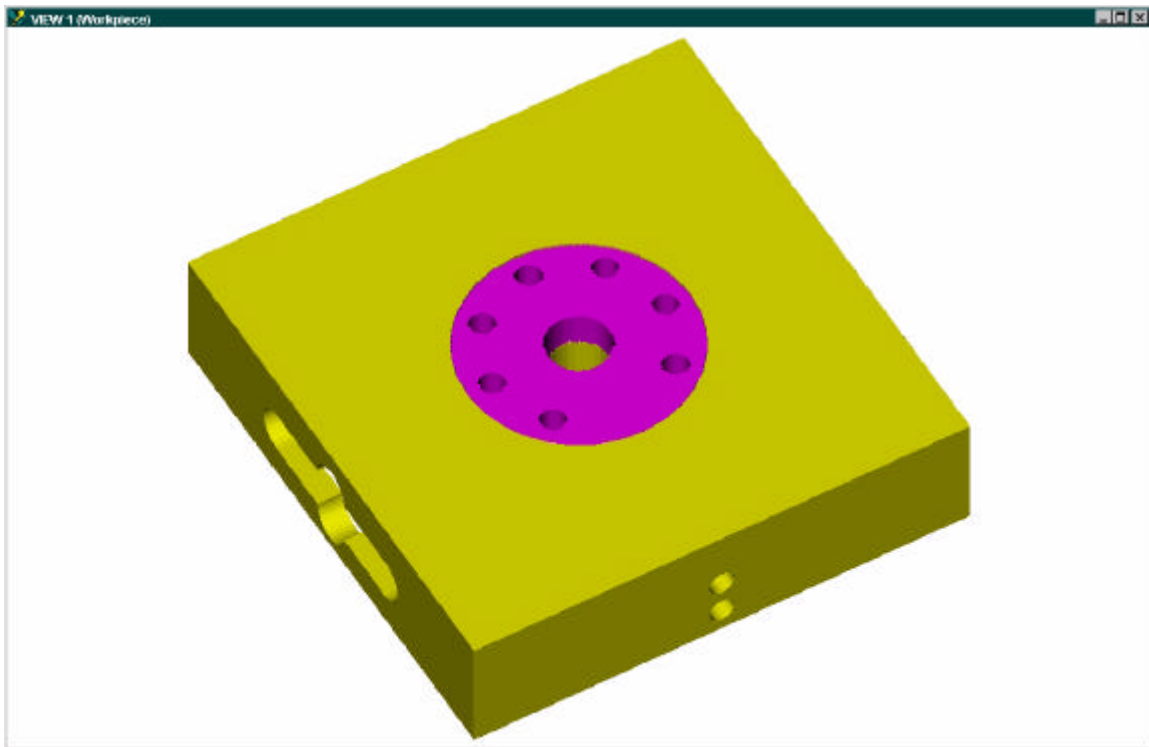
Yes, there are several places where excess material exists.

4 small standing posts are present because design model was created with sharp corners instead of modeling the corner radii that are machined into the part- this is most likely not a problem.

2 larger standing posts caused by the cutter which reduced the corners and did not cut in these corners- this most likely IS a problem.

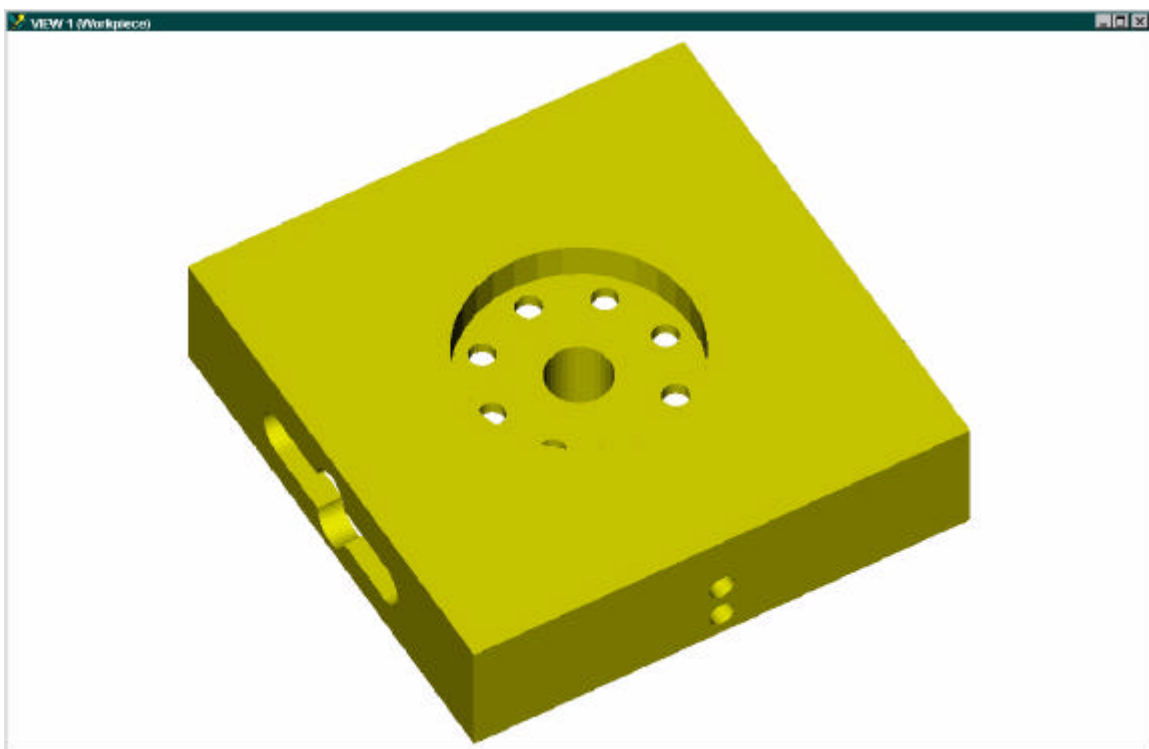
Large disk of material is present because design model has a cavity on the bottom that has not yet been machined- this will not be a problem if an operation follows to machine the cavity.

Click on **Design Display**



**Comparison Type=Gouge
Compare**

Notice the cavity in bottom of Design model



Session 13- Control the Tool Path Simulation

This session shows how to control the simulation of an APT tool path file. Full control is provided over the interactive simulation enabling users to interrupt the simulation at any time by pressing

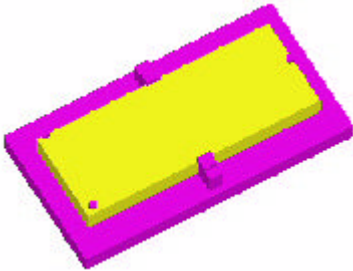


Stop or <Ctrl>+<C>. In addition, VERICUT can be configured to stop tool path processing based on a variety of conditions, including: when errors are detected, after tool changes, after a specified number of cuts have been performed, etc. This session demonstrates practical use of VERICUT simulation controls.

Session Steps:

1. Open the "rib.usr" User file

File menu > Open
Shortcut=CGTECH_SAMPLES
File Name=rib.usr > OK > OK
If a message appears, press **Reset**.



2. Open the Status window to monitor the machining operation

Info > Status

3. Configure VERICUT to use only the cutting portion of the tool during the simulation; cutter descriptions are to come from "CUTTER" records in the tool path file

Setup > Toolpath
Un-check Use Holders
OK

4. Configure VERICUT to stop after tool changes and when an error is detected

Setup > Motion
Stop At=Tool Change
Check Stop At Maximum Errors
Stop At Maximum Errors=1

5. Configure VERICUT to issue an error if material is cutusing a feedrate of 100 or higher

Fast Feed=100

Apply

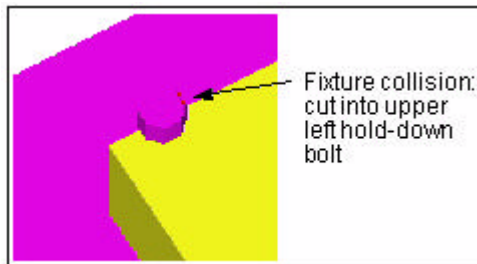
OK

6. Cut the model - why does VERICUT stop?

Play to end  ->tool change with record 20: CUTTER/1.25,19,0,0,0,0,2.5 (see Status window)

7. Cut the model - why does VERICUT stop?

Play to end  ->"Collision with clamp or fixture occurred at record sequence number 134"
(see VERICUT window message area)

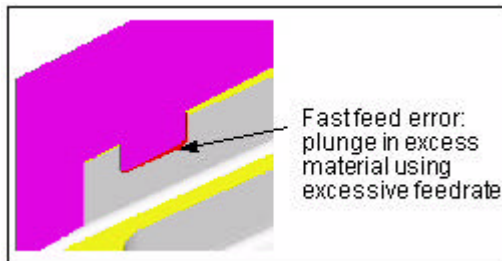


8. Cut the model - why does VERICUT stop?

Play to end  ->tool change record 592: CUTTER/1.25,19,0,0,0,0,2.5

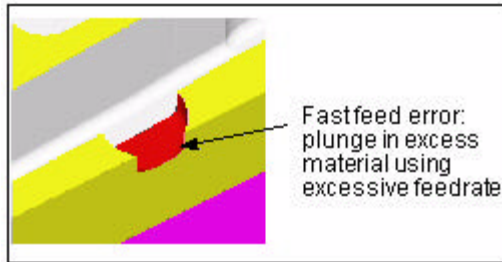
9. Cut the model - why does VERICUT stop?

Play to end  ->"Fast feed rate removed material at record sequence number 703"



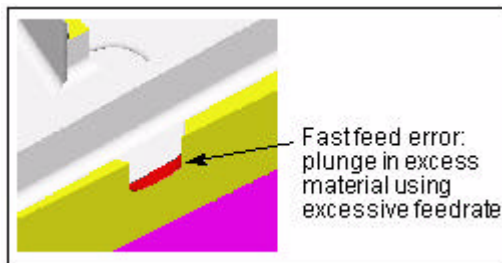
10. Cut the model - why does VERICUT stop?

Play to end  ->"Fast feed rate removed material at record sequence number 856"



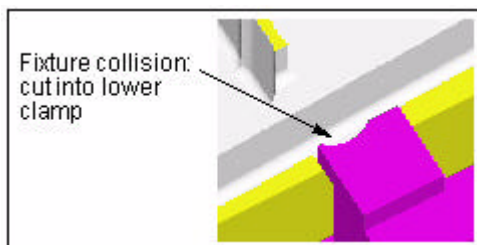
11. Cut the model - why does VERICUT stop?

Play to end  -> "Fast feed rate removed material at record sequence number 1030"



12. Cut the model - why does VERICUT stop?

Play to end  -> "Collision with clamp or fixture occurred at record sequence number 1040"



13. Configure VERICUT to continue processing without stopping on errors

Setup > Motion
Un-Check **Stop At Maximum Errors**
OK

14. Cut the model - why does VERICUT stop?

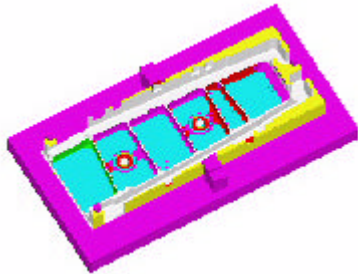
Play to end  -> tool change record 1441: CUTTER/1.25,.12,0,0,0,0,2.5

15. Configure VERICUT to continue process the rest of the tool path file without interruptions, then cut the model

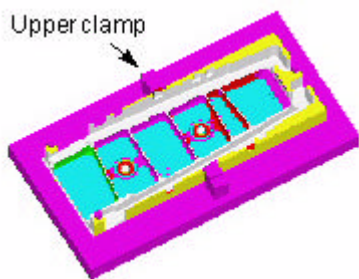
Setup > Motion
Stop At = File End
OK

16. Cut the model - how many errors did VERICUT detect?

Play to end  ->VERICUT detected over 300 errors! Without VERICUT this NC programmer would be looking for a new occupation.



17. Use Feature/History to identify the tool path record responsible for cutting the upper clamp (see below)



Analysis > X-Caliper
Feature/History Tab
click on machine cut in upper clamp -> Record Number: 1044

18. Rewind the tool path and configure VERICUT to start processing 1 record prior to the error record identified during the preceding above

Rewind Toolpath 
Setup > Motion
Start At Line=1043
OK

19. Step (single block) through motions that cut the clamp

Step 
Step (or pressing the Space Bar will repeat **Step**)
Step

Session 14- Command Line Options & Batch Processing

This session shows how to run VERICUT in "batch" mode where VERICUT runs in the background. Batch processing frees the user to perform other tasks (such as file system maintenance, additional programming tasks, etc.) while VERICUT continues to be productive as well.

Assuming that a User or In-process file has been saved containing the essentials for tool path simulation (stock, tool path, tools), it is relatively simple to configure VERICUT to output the desired data during a batch process.

This session demonstrates capturing snapshots of errors detected during the batch tool path simulation session and saving the resulting cut model in an In-process file.

Session Steps:

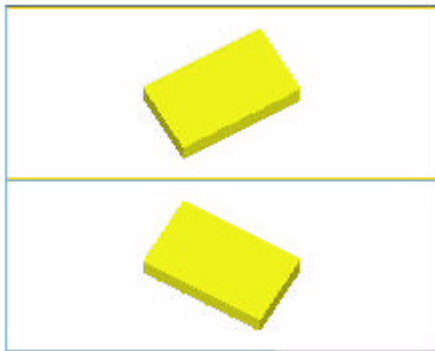
1. In VERICUT, open the "noserib.usr" User file

File > Open

Shortcut: CGTECH_SAMPLES

File Name=noserib.usr > OK > OK

If a message appears, press **Reset**.



2. Configure to save snapshots to an Image file with basefile name "batch.img" each time an error is detected

File > Images > Record

Snapshot Events=Error > Animation Control=Disabled >

Filename=batch.img

Record > Stop > Append > Close

3. Configure to save an In-process file named "batch.ip" after the tool path has been simulated

File menu > Auto Save

under "In Process" toggle **AutoSave=File End, File=batch.ip**

OK

4. Save a log file with a different file name "batch.log"

Info > VERICUT Log

File > Reset Log File > Yes

File > Save As

Shortcut=Working Directory

File Name=batch.log

5. Save a new User File named "batch.usr"

File > Save As

Shortcut=Working Directory

File Name=batch.usr

Save

File > Exit

6. Use the CGTech Software Manager to create a VERICUT batchsession for the "batch.usr" User file

From the Taskbar, click **Start > Programs > CGTech VERICUT 5.0 >**

BatchWizard

Select **"Start a new session" > Next**

Add > Command= Vericut > Next

Add the "usr=batch.usr" command line option:

Under **"Select Options"** select **"usr=<file>"** -> automatically enters **"usr= "** in Selection field

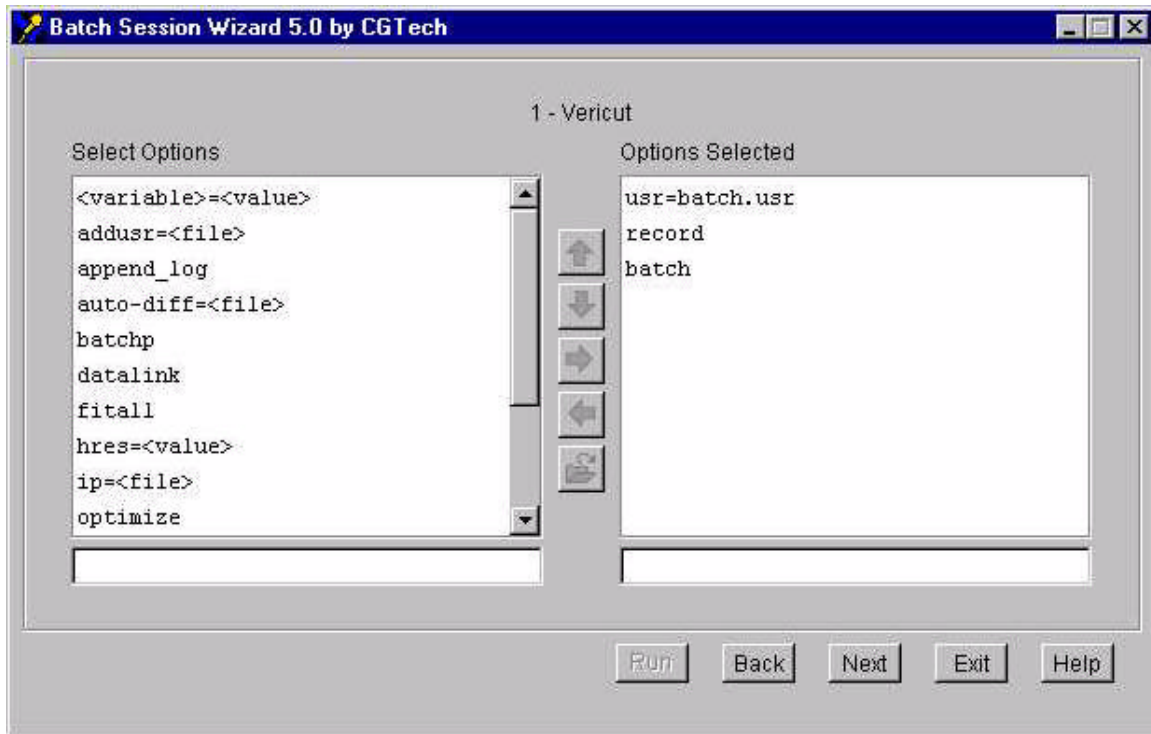
Press **Browse**  to open a file selection window, then select **"batch.usr"** -> moves **"usr=batch.usr"** command line option under **"OptionsSelected"**

Add the **"record"** command line option:

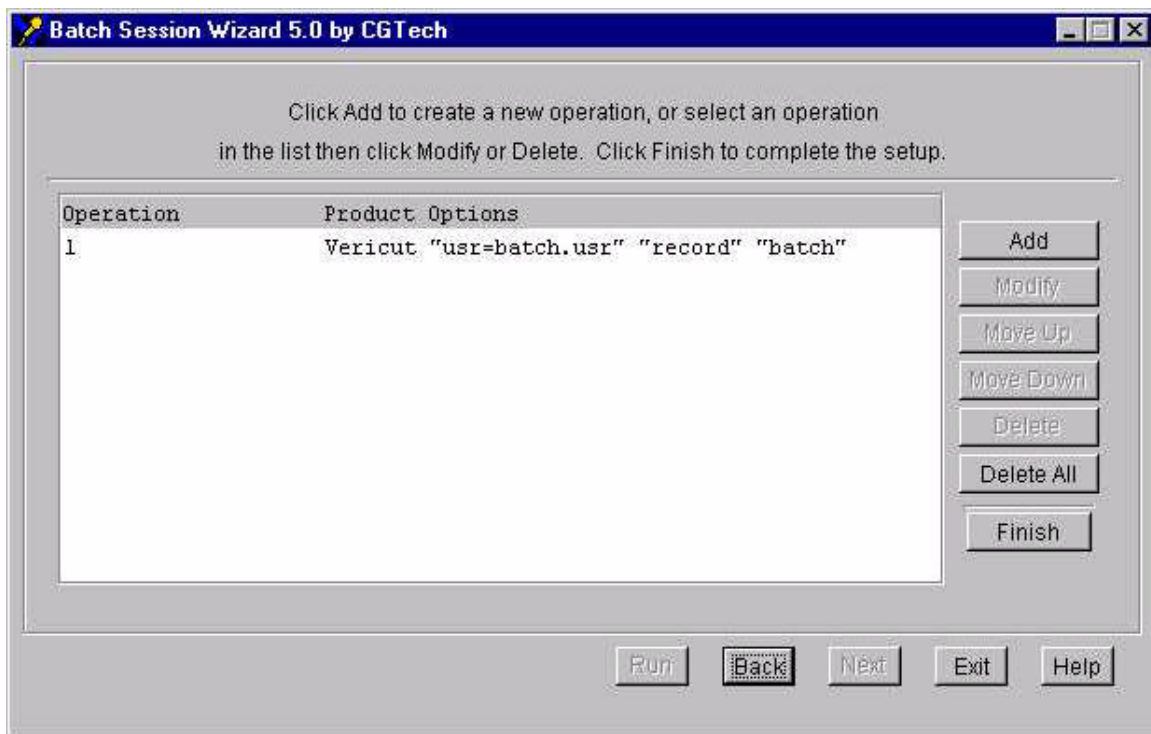
Under **"Select Options"** select **"record"**, press 

Add the **"batch"** command line option:

Under **"Select Options"** click on **"batch"**, "press  to move **"batch"** command line option under **"Options Selected"**



Next



NOTE: to see the entire batch operation line it may be necessary to stretch the Batch Wizard window left or right.

Finish

Selection=batch.vcs > OK -> saves a VERICUT BatchSession file

Batch Session files are platform independent. They can be created on any supported computer platform,

then run on any other supported computer platform.

7. run the "batch.vcs" Batch Session file

Select **"Run an existing session"** > **Next**

Select

Shortcut=Working Directory

File Name=batch.vcs > **OK**

Run

Windows NOTE: a DOS window may appear indicating the batch session is running. DO NOT close this window as the batch process will prematurely terminate. VERICUT will automatically close the window when the batch session is done.

8. After the batch session is done, execute VERICUT, then open the "batch.ip" In-process file

File > In Process > Open > Filename=batch.ip



8. Play back the "batch.img" Image file with 2 second delay between snapshots

File menu, Images>Playback > Snapshot Delay=2 > Filename=batch.img>

Load > Play (solid green arrow pointing right)

Note that there were 4 snapshots recorded: 1-3 recorded for Holder collisions, 4- final part image. Consult the **VERICUT Log** to see which tool path records caused the errors.

Session 15- Model Export

This session shows how to export a cut model into an IGES format, then read back that file in VERICUT and use it as a stock for subsequent operations.

1. Open a user file.

File Open

Shortcut:CGTECH_SAMPLES

File Name = sprout.usr **OK OK**

If a message appears, press **Reset**.

2. Delete sprout3.tp from Toolpath list

Setup Toolpath

Select sprout 3.tp Delete

Remove all 4 sprout3.tp

3. Cut the workpiece

Play to end



4. Export the cut model into an IGES file

File Export Model

File = sprout.igs

File Type = IGES

Model Coordinates = Stock

Process = Machined Features

Output

Close

Note: it may take a while.

Suggestions on discussions subjects: - Sport - High speed machining - Cars - " who has the fastest CPU?"

5. Reset

Reset



6. Delete the stock block

Model Component Tree

Select the stock **block**

Delete

7. Load the IGES file as a stock

Model Model Definition Import

IGES File=sprout.iges

Tolerance=.010

8. Cut the rest of the part

Setup Toolpath

Clear

Add sprout3.tp orientation3

Add sprout3.tp orientation4

Add sprout3.tp orientation5

Add sprout3.tp orientation6

Apply

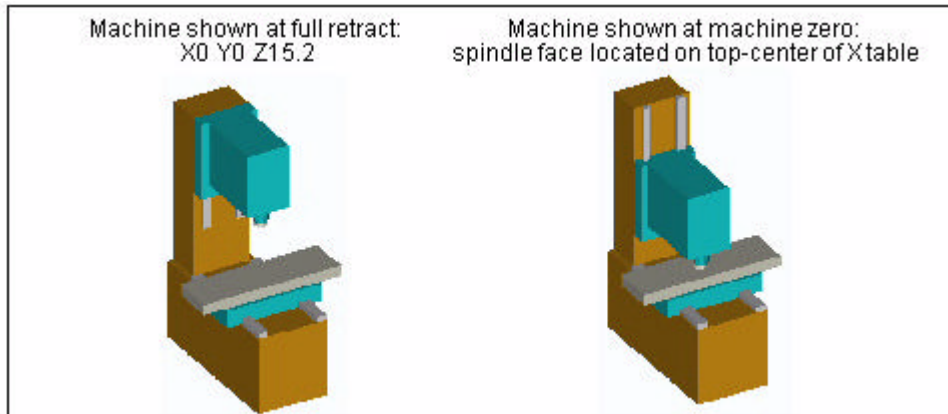
Close

Play to end



Session 16- Configure a User file to Simulate G-codes

This session shows how to configure VERICUT for processing a G-code tool path file. The "3axtltip.mcd" G-code tool path file to be simulated has been programmed to drive the tool tip. The tool path is destined to be run on a 3-axis vertical mill similar to the one shown below. A Mazak Mazatrol M-32 CNC milling control interprets the G-codes for this milling machine.



Sample 3-axis Vertical Mill

The Machine Simulation system provides many sample machine and control configurations which users can choose to simulate their NC machining environment. This session demonstrates how to create a User file for a specific NC machine and control combination for simulating the G-code tool path.

The basic requirements for G-code tool path simulation are the same as for any other tool path type (stock, tool path, and cutting tool descriptions), however, there are additional considerations:

- Tool descriptions are not typically present in a G-code tool path file. A Tool Library is typically used to supply cutter descriptions.
- VERICUT must have knowledge of the NC machine kinematics & control capabilities, as well as job-related data such as: the initial machine location prior to G-code processing, offset register values entered at the NC control, etc.
- With the Machine Simulation system, this information is stored as follows:
 - kinematic properties of an NC machine are stored in a Machine file
 - information about the capabilities and methods of how an NC control interprets G-codes is stored in a Control file
 - job-related data, as well as the names of the above mentioned Machine and Control files are stored in the User file

Session Steps:

1. Start from a new Inch User file

- File > Properties
- Default units=Inch > OK
- File > New

3. Add a Stock model to be a 5 x 6 x 2 inch block

- Model > Model Definition: Model tab
- Type: Block

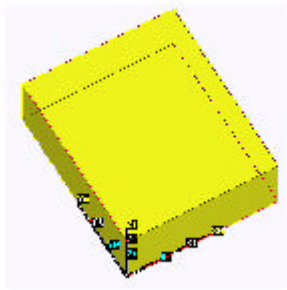
Length(X)=5

Width(Y)=6

Height(Z)=2

Add

Fit 



4. Specify the "3axtltip.mcd" G-code tool path file to be simulated and set the proper Toolpath Type

Setup > Toolpath

Toolpath Type=G-Code Data

Add

Shortcut=CGTECH_SAMPLES

File Name=3axtltip.mcd

5. Use the "\${CGTECH_RP2LIB}g3vmitt.mch" generic 3-axis mill with "\${CGTECH_RP2LIB}mazm32.ctl" Mazak Mazatrol M-32 CNC milling control

Setup > Machine > Open

Shortcut=CGTECH_LIBRARY

File Name=g3vmitt.mch

Setup > Control > Open

Shortcut=CGTECH_LIBRARY

File Name=mazm32.ctl

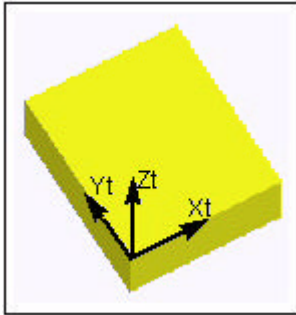
6. Specify that "tool tip" programming is used for this toolpath

Setup > G-Code > Settings > Settings Tab

Programming Method=Tool Tip

OK

7. Orient the tool path to be setup on the top left corner of the stock as shown below



Setup Required for Tool Path "3axtltip.mcd"

Setup > Coordinate Systems

Position=0 0 2 (or select the top left corner as shown above)

The "3axtltip.mcd" tool path contains "T" words which specify the tool number of the cutters used to machine the part. The "3axtltip.tls" Tool Library file contains cutter descriptions that correspond to the tool numbers referenced in this G-code tool path.

8. Configure to retrieve tool data from the "3axtltip.tls" Tool Libraryfile

Setup > Tool Manager

File > Open

Shortcut=CGTECH_SAMPLES

File Name=3axtltip.tls

Open

File > Close > Yes

9. Cut the model

Play to End 



Session 17- Use OptiPath Manager to Create an OptiPath Library

This session shows how to use the OptiPath Manager function to define the OptiPath records required to optimize cutting in H13 toolsteel (approx. 200 HB). Once defined, the OptiPath records are stored in an OptiPath Library file. The following session (Optimize Tool Path Feedrates via OptiPath tool list method) demonstrates how to configure VERICUT for optimizing a G-code tool path file, including using the OptiPath Library created during this session.

Session Steps:

Define OptiPath Records

1. Start from a new Inch User file

File > Properties

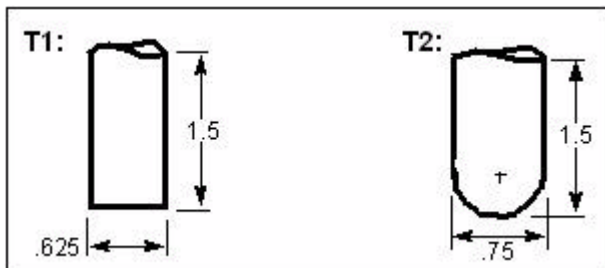
Default units=Inch > OK

File > New

2. Access the OptiPath Manager

OptiPath > OptiPath Manager

Optimization settings will be established differently to accommodate the different cutting performed by each tool (see below). The part will be cut on a 3ax vertical mill in a rigid setup.



Cutters Used by Tool Path "op_mold.mcd"

Tool 1 (T1):

Description: .625 dia. 4 flute carbide end mill

Operation: planar milling not to exceed .5" depth, 1200 RPM

Feedrates for this tool will be established from a known successful cutting condition: assume the cutter is successful cutting full width in .3" depth passes at 8 IPM feedrate. The feedrate used to enter material should not be more than 8 IPM.

Tool 2 (T2):

Description: .75 dia. 2 flute Ingersoll carbide end mill

Operation: semi-finish profile milling

The light cuts performed by this tool will be optimized by constant volume combined with chip thickness. This method of optimization varies feedrates based on the volume of material removed. Assume volume removal rates are not known for this tool; OptiPath record will be copied from the Ingersoll OptiPath Library.

Configure Optimization Settings for the .625 dia. Flat End Mill:

3. Identify a new OptiPath record to be added for the .625dia. flat end mill as follows:

Stock material to be cut= H13 Tool Steel
Machine that will cut the part= 3ax Mill
Tool description= .625D 1.5H FEM, Carbide
#Teeth= 4
Add
Material=H13 Tool Steel
Enter
Machine=3ax Mill
Enter
Tool Description=.625D 1.5H FEM, Carbide
Enter
Teeth=4
Enter
Specify Temporary Cutter Shape



Flat Bottom End Mill
Diameter(D)=.625
Height(H)=1.5
OK

4. Adjust to a known successful cutting condition

Axial Depth=.3 (Value can be adjusted with the slide bar or typed)
Radial Width=.625
Feed Per Minute=8
Spindle Speed=1200
Select **Spindle Speed** (When checked, OptiPath will override the existing Spindle speed with this value)

Computation shows calculated volume removal rate= 1.5 cubic in. per min.

5. Configure OptiPath settings to "Optimize by Constant Volume" and add more cuts when needed

Click on **Volume Removal**
Settings Tab > Add More Cuts
Apply

6. On the Optimization Settings Tab specify settings for the following conditions:

Settings Tab

Un-Select all **Default**

Minimum Feedrate Change=3 (controls how often the optimized feedrate is changed)

Clean-up Feedrate=80 (spring pass)

Minimum Cutting Feedrate=2

Maximum Cutting Feedrate=80

Feed/Speed Tab

Un-Select **Default** for **Air Cut Feed Rate**

Air Cut Feed Rate=150

7. Use Entry/Exit to use 8 IPM feedrate when entering and exiting material: from .1 before contacting material to .1 into material, then apply the new optimization settings

Entry/Exit Tab

Entry Feedrate=Feed/Minute : 8

Clearance Distance=.1

Cut Distance=.1

Exit Feedrate=Feed/Minute : 8

Clearance Distance=.1

Cut Distance=.1

Apply

8. Test how optimization will be performed under various cutting conditions

Testing ensures that optimization will be performed as expected. If unsatisfactory results are experienced, adjust optimization settings and re-test until satisfied.

Feed/Speed Tab

Enter various cut depths and widths

Example:

Axial Depth=.1

(Feed per Minute= 24)

Radial Width=.5

(Feed per Minute= 30)

Configure Optimization Settings for the .750 dia. Ball End Mill:

9. Add a new OptiPath record for the .750 dia. ball endmill, edit the new record name as follows:

Stock material to be cut= H13 Tool Steel

Machine that will cut the part= 3ax Mill

Tool description= .750D 1.5H BEM, Carbide

#Teeth= 4

Add

Material=H13 Tool Steel

Machine=3ax Mill

Tool Description =.750D 1.5H BEM, Carbide

Teeth=4

Specify Temporary Cutter Shape



Ball End Mill

Diameter(D)=.75

Height(H)=1.5

OK

10. Adjust to a known successful cutting condition

Axial Depth=1 (Value can be adjusted with the slider or typed)

Radial Width=.125

Feed Per Minute=12

Spindle Speed=1200

Select **Spindle Speed** (When checked, OptiPath will override the existing Spindle speed with this value)

11. Change the OptiPath settings for this tool to optimize by "Constant Volume" and "Constant Chip Thickness" (continue to add more cuts when needed)

The following settings were copied from the previously defined OptiPath record and will be used as is:

Minimum Feedrate Change= 3

Air cut feedrate= 150

Clean-up feedrate= 80

Maximum cutting feedrate= 80

Minimum cutting feedrate= 2

Entry/exit feedrate= 8

Click on **Volume Removal**

Click on **Chip Thickness**

Settings Tab > Add More Cuts

Apply

12. On the Optimization Settings Tab specify settings for the following conditions:

Settings Tab

Un-Select all **Default**

Minimum Feedrate Change=3 (controls how often the optimized feedrate is changed)

Clean-up Feedrate=80 (spring pass)

Minimum Cutting Feedrate=2

Maximum Cutting Feedrate=80

Feed/Speed Tab

Un-Select **Default** for **Air Cut Feed Rate**

Air Cut Feed Rate=150

13. If desired, test how optimization will be performed under various cutting conditions

14. After test results are satisfactory, add the new OptiPath record to the library

Apply

Save the New OptiPath Library:

15. Save the OptiPath records in an OptiPath Library file named "optipath.olb" and close the OptiPath Manager

File > Save As

Shortcut=Working Directory

File Name=Session14.olb

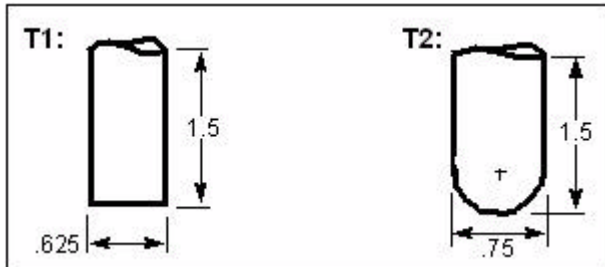
Save

Session 18- Optimize Tool Path Feedrates via Tool Library method

This session shows how to configure VERICUT for optimizing a G-code tool path file by adding references to OptiPath records to tools stored in a previously defined Tool Library file. The sample G-code toolpath file to be optimized uses 2 cutting tools: T1, T2. The tools have been defined and stored in the Tool Library file that will receive the OptiPath record references. The demonstration shows how to link cutting tools in the Tool Library file with OptiPath records in an OptiPath Library file, as well as optimize the tool path file motions and review the optimized tool path file. See also: Session 19- Optimize Tool Path Feedrates via OptiPath tool list method

Session Steps:

Optimize an Inch Tool Path



Cutters Used by Tool Path "op_mold.mcd"

1. In VERICUT, open the "op_mold.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=op_mold.usr > OK > OK

If a message appears, press **Reset**



2. Use the Tool Manager to link OptiPath records to cutting tools used by the tool path file, then save a new "optipath.tls" Tool Library file

Tool 1:

Setup > Tool Manager...

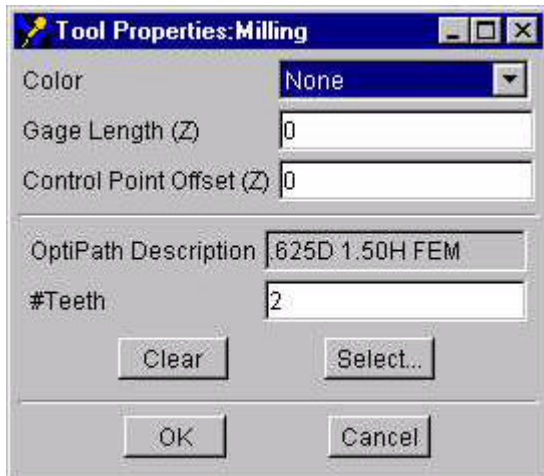
In the tool list, highlight tool ID 1 - .625D 1.50H FEM

Modify

Properties > Select

Highlight **Stock Material**=H13 Tool Steel > **Machine**=3axMill ,

Tool Description (#Teeth)=.625D 1.5HFEM, Carbide (4) > **OK** (NOTE: the window may require stretching to see all information)



Tool Properties window: **OK**, then **Tool Modify** window: **OK**

Tool 2:

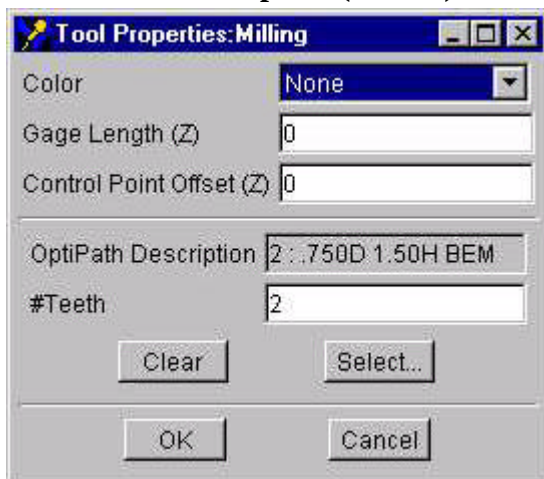
Setup > Tool Manager

Highlight tool ID 2 - .750D 1.50H BEM > **Modify**.

Properties > Select...

Highlight **Stock Material**=H13 Tool > **Machine**=3ax Mill ,

Tool Description (#Teeth)=.750D 1.5HBEM, Carbide (4) > **OK**



Tool Properties window: **OK**, then Tool Modify window: **OK**

Tool Manager window: **File > Save As**

Shortcut=Working Directory

File Name=optipath.tls

Save

File > Close > Yes

3. Use OptiPath Control to specify the "optipath.olb" OptiPathLibrary file is to be used to create an optimized tool path named "op_mold.mco"

```
OptiPath > Control
OptiPath Library=optipath.olb
(If "optipath.olb" is not available use "{CGTECH_VCSAMPLES}op_mold.olb")
Optimized Toolpath=op_mold.mco
```

4. Choose "Setup..." to specify that cuts in H13 tool steel on the 3ax mill are to be optimized, then close that window

```
Material=H13 Tool Steel
Machine=3ax Mill
```

5. Turn OptiPath on to start toolpath optimization

```
On
OK
```

6. Open the Status window and configure to monitor feedrates calculated during optimization

```
Info > Status
Configure
Toggle Graph
Toggle OP Time
OK
```

During processing the Feedrate field displays the programmed feedrates while the OP Feedrate field displays the feedrates optimized by OptiPath.

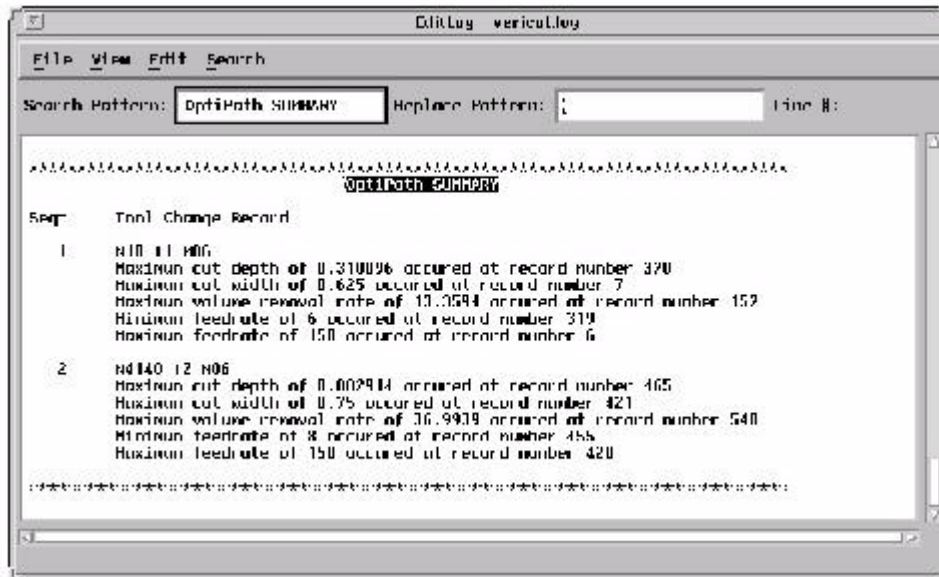
7. Cut the model



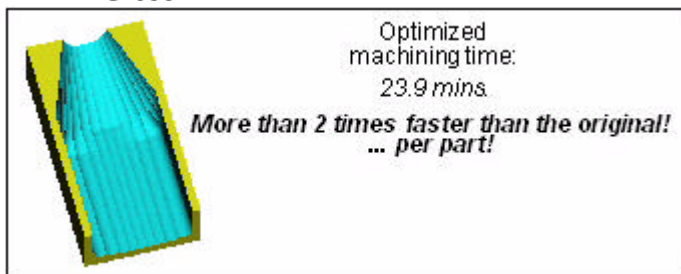
NOTE: The OptiPath light is turned red, indicating that the feedrates are being optimized.

8. Open the Log file window and review the "OptiPath SUMMARY", then close the Log file viewer

```
Info > VERICUT Log...
Scroll to bottom of file and search for the "OptiPath SUMMARY" header.
```



Sample Log File OptiPath Summary
Close



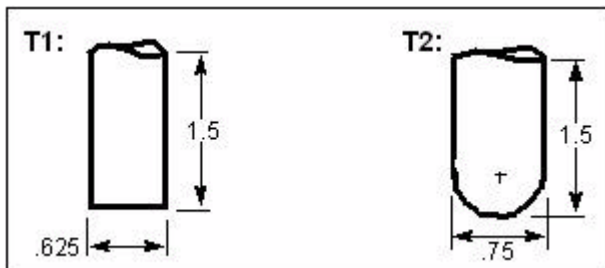
Session 19- Optimize Tool Path Feedrates via OptiPath tool list method

This session shows how to configure VERICUT for optimizing a G-code tool path file by building a Tool List to reference OptiPath records stored in a previously defined OptiPath Library file. The sample G-code tool path file to be optimized uses 2 cutting tools: T1, T2. The tools have been defined and stored in a Tool Library file. The demonstration shows how to link cutting tools in the tool path file with OptiPath records in an OptiPath Library file, as well as optimize the tool path file motions and review the optimized tool path file.

See also: Session 18- Optimize Tool Path Feedrates via Tool Library method

Session Steps:

Optimize an Inch Tool Path



Cutters Used by Tool Path "op_mold.mcd"

1. In VERICUT, open the "op_mold.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=op_mold.usr > OK > OK

If a message appears, press **Reset.**, OK



2. Use OptiPath Control to specify the "optipath.olb" OptiPathLibrary file is to be used to create an optimized tool path named "op_mold.mco"

OptiPath > Control

OptiPath Library=optipath.olb

(If "optipath.olb" is not available use "{CGTECH_VCSAMPLES}op_mold.olb")

Optimized Toolpath=op_mold.mco

OK

3. Link cutting tools in the tool path file to OptiPath records as follows

A tool list is automatically generated by scanning the tool path file. The tool change records represent cutting tools used by the tool path file. It is these records that must be linked to OptiPath records for use during tool path optimization, as described by the next step.

Setup > Toolpath

Tool Change By=List

Use Tool list

Make sure **Prompt for Optipath...** is **NOT** selected

Build Tool List --> scans the tool path and generates the tool list

Under **OptiPath Settings**

Tool 1 (T1) use OptiPath record= .625D 1.5H FEM, Carbide (4)

Tool 2 (T2) use OptiPath record= .750D 1.5H BEM, Carbide (4)

OK

4. Turn OptiPath on to start tool path optimization

OptiPath > Control

Material=H13 Tool Steel

Machine=3ax Mill

On

OK

5. Open the Status window and configure to monitor feedrates calculated during optimization

Info > Status

Configure

Toggle Graph

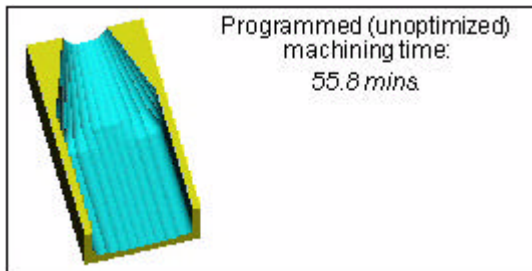
Toggle OP Time

OK

During processing the Feedrate field displays the programmed feedrates while the OP Feedrate field displays the feedrates optimized by OptiPath.

6. Cut the model

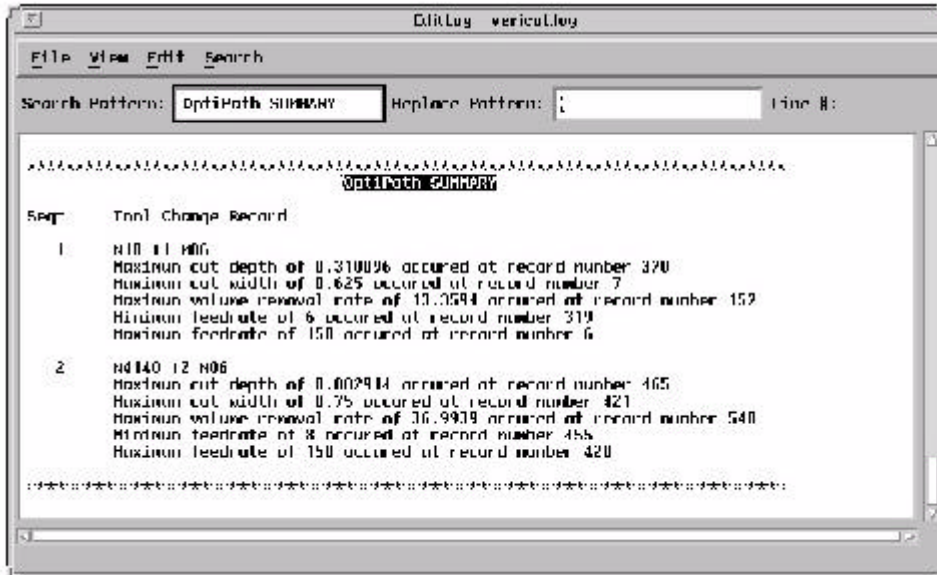
Play to end



7. Open the Log file window and review the "OptiPath SUMMARY", then close the Log file viewer

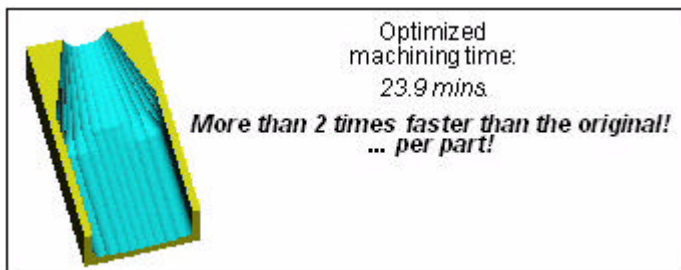
Info > VERICUT Log...

Scroll to bottom of file and search for the "OptiPath SUMMARY" header.



Sample Log File OptiPath Summary

Close



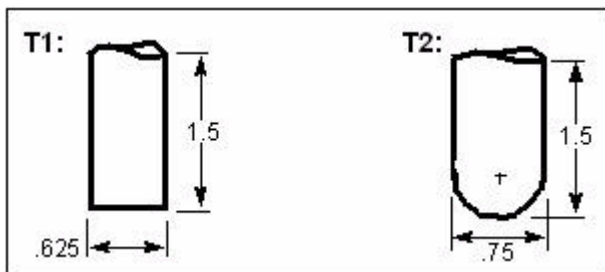
Session 20- Build/Modify an OptiPath library while building a tool list

This session shows how to build/modify an OptiPath library when building a Tool List. The sample G-code tool path file to be optimized uses 2 cutting tools: T1, T2. The demonstration shows how to link cutting tools in the tool path file with existing OptiPath records or create new ones interactively while VERICUT is scanning for the tool change records.

See also: Session 19- Optimize Tool Path Feedrates via OptiPath tool list method

Session Steps:

Optimize an Inch Tool Path



Cutters Used by Tool Path "op_mold.mcd"

1. In VERICUT, open the "op_mold.usr" User file

File > Open

Shortcut=CGTECH_SAMPLES

File Name=op_mold.usr > OK > OK

If a message appears, press **Reset**



2. Use OptiPath Control to specify the "optipath.olb" OptiPathLibrary file is to be used to create an optimized tool path named "op_mold.mco"

OptiPath > Control

OptiPath Library=op_mold.olb

Optimized Toolpath=op_mold.mco

Material=H13 Tool Steel

Machine=3ax Mill

OK

3. Link cutting tools in the tool path file to OptiPath records as follows

A tool list is automatically generated by scanning the tool path file. The tool change records represent cutting tools used by the tool path file. By checking the **Prompt for Optipath**, at every tool change the OptiPathManager will pop-up.

Setup > Toolpath

Tool Change By=List

Use Tool list

Check **Prompt for Optipath**

Build Tool List --> scans the tool path and generates the tool list

In the Optimization Settings window :

Select Existing > Select .625D 1.50H FEM,... > OK

OK (an existing OptiPath Record is now selected for that tool)

The OptiPath Manager pops up for the next tool

Change the **Feed Per Minute=25**

Select **Volume Removal**

Select **Chip Thickness**

OK

A new OptiPath Record have been created

OK > OK

4. Turn OptiPath on to start tool path optimization

OptiPath > Control

On

OK

5. Open the Status window and configure to monitor feedrates calculated during optimization

Info > Status

Configure

Toggle **Graph**

Toggle **OP Time**

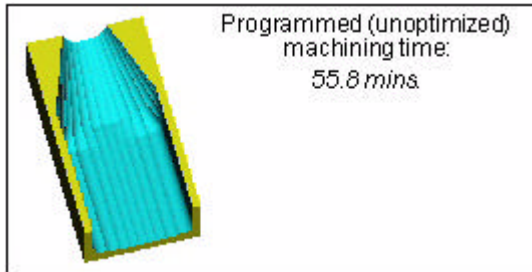
OK

During processing the Feedrate field displays the programmed feedrates while the OP Feedrate field displays the feedrates optimized by OptiPath.

6. Cut the model

Play to End

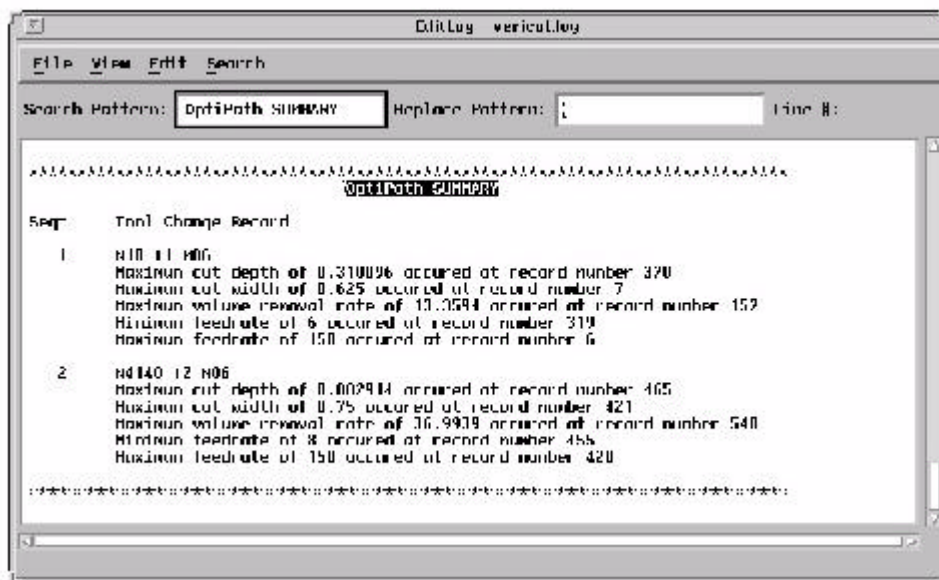




7. Open the Log file window and review the "OptiPath SUMMARY", then close the Log file viewer

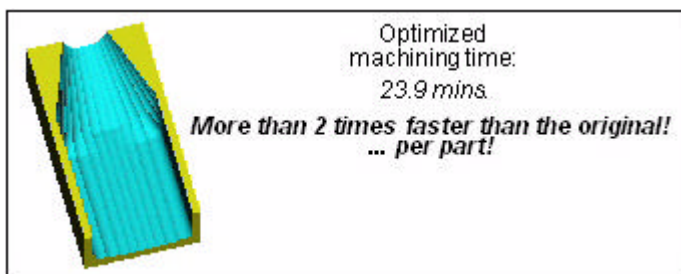
Info > VERICUT Log...

Scroll to bottom of file and search for the "OptiPath SUMMARY" header.



Sample Log File OptiPath Summary

Close



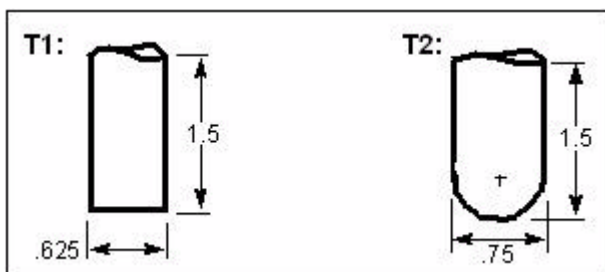
Session 21- Run OptiPath interactively

This session shows how to build/modify an OptiPath library interactively while cutting the part. The sample G-code tool path file to be optimized uses 2 cutting tools: T1, T2. The demonstration shows how to link cutting tools in the tool path file with existing OptiPath records or create new ones interactively while VERICUT cutting the part. At the end of the session a results file is generated, showing each tool used, the OptiPath settings and time saving for each tool.

See also: Session 20- Build/Modify an OptiPath library while building a tool list

Session Steps:

Optimize an Inch Tool Path



Cutters Used by Tool Path "op_mold.mcd"

1. In VERICUT, open the "op_mold.usr" User file

File > Open

Shortcut: CGTECH_SAMPLES

File Name=op_mold.usr > OK > OK

If a message appears, press **Reset**



2. Use OptiPath Control to specify the "optipath.olb" OptiPathLibrary file is to be used to create an optimized tool path named "op_mold.mco"

OptiPath > Control

OptiPath Library=op_mold.olb

Optimized Toolpath=op_mold.mco

Material=H13 Tool Steel

Machine=3ax Mill

Interactive

OK

3. Open the Status window and configure to monitor feedratescalculated during optimization

Info > Status

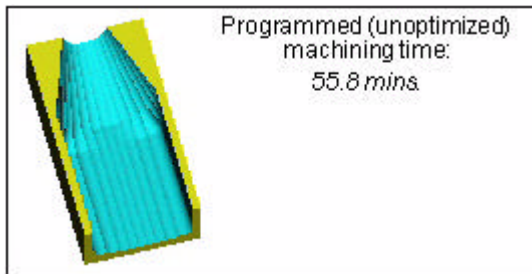
Configure

Toggle Graph

Toggle OP Time

OK

During processing the Feedrate field displays the programmedfeedrates while the OP Feedrate field displays the feedrates optimizedby OptiPath.



4. Cut the model

Play to End 

At every tool change the OptiPath Managerwill pop-up

In the Optimization Settings window :

Select Existing > Select .625D 1.50H FEM,... > OK

OK (an existing OptiPath Record is now selected for thattool)

The OptiPath Manager pops up for the next tool

Change the **Feed Per Minute=25**

Select **Volume Removal**

Select **Chip Thickness**

OK

A new OptiPath Record have been created

5. Generate a HTML report, Results File

Info > Results File

Scroll to bottom of file and search for the "OptiPath SUMMARY" header.

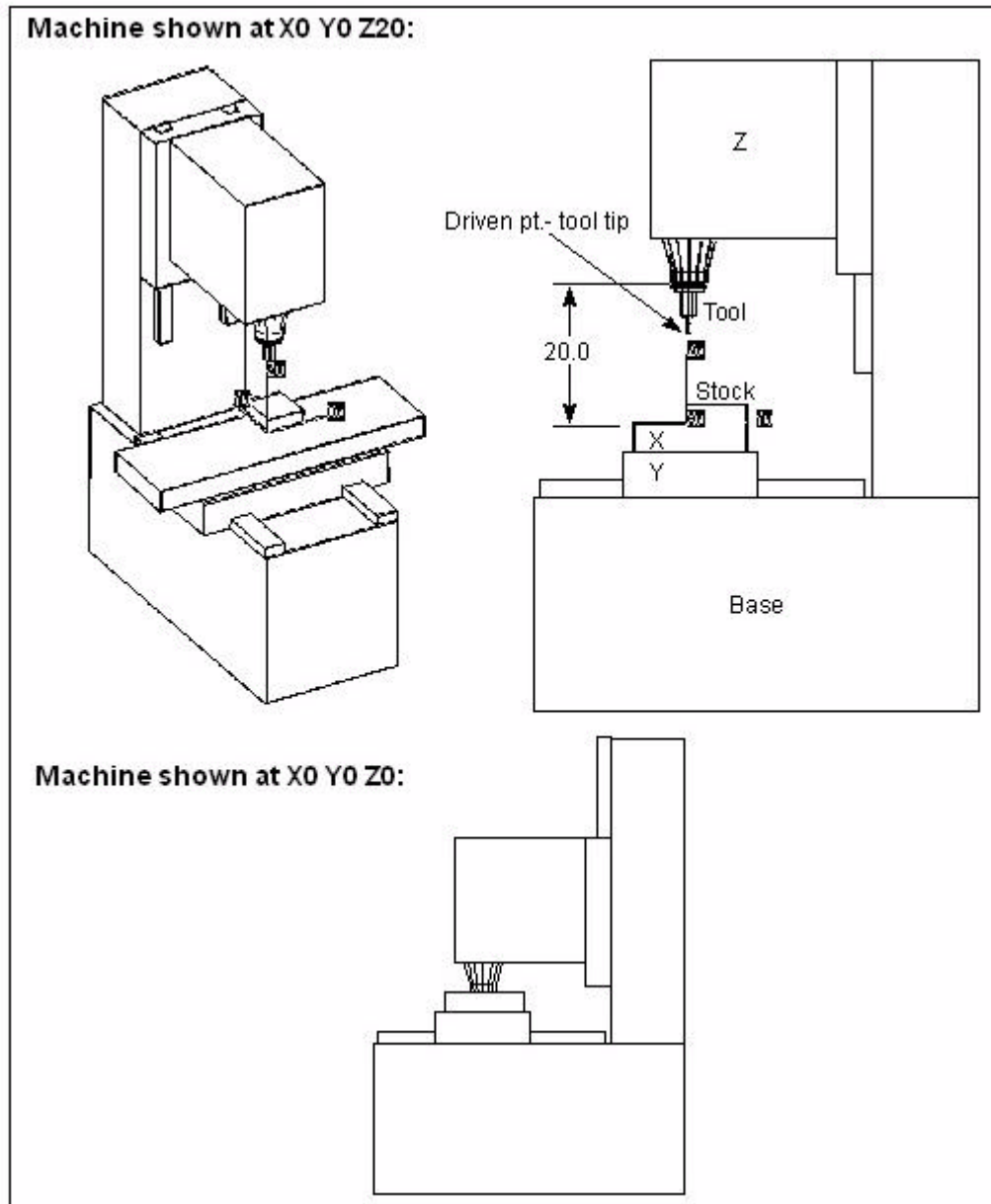
Tool Summary									
	Seq.	Record	Tool Description	Height	OptiPath Record	Optimize By	Original Time	Optimized Time	Difference
	1	3: N10 T1 M06	1: 625D 1.50H FEM	1.5	625D 1.50H FEM, Carbide #4	Tables	40.6807	21.9898	45.9 %
	2	417: N4140 T2 M06	2: 750D 1.50H BEM	1.5	2: 750D 1.50H BEM #2	Chip Thickness: 0.0049, Constant Volume: 6.88	15.1377	8.10405	46.5 %
Total Time							55.8184	30.0938	46.1 %
Number of Errors : 0									
Number of Warnings : 0									
Total cycle time since last rewind : 55.8184									

Sample Results file Tool Summary

Close

Session 22- Build a Kinematic Model of a 3-axis Mill

This session shows how to configure a VERICUT Machinefile that describes the kinematics of a 3-axis vertical mill, then configure VERICUT to simulate a G-code tool path destined for the NC machine. The 3-axis mill machine is controlled via a Yasnac control and programmed using the "tool tip" programming method.



3-axis Vertical Mill to Run "gtooltip.mcd "

Session Steps:

1. Start from a new Inch User file

File > Properties
Default units=Inch > OK
File > New

2. Display the model, the Component and the Machine axissystems

View > Axes
Select Model
Select Component
Select Machine Origin
Close

3. Open the Yasnac MX-3 Control file for mill machines

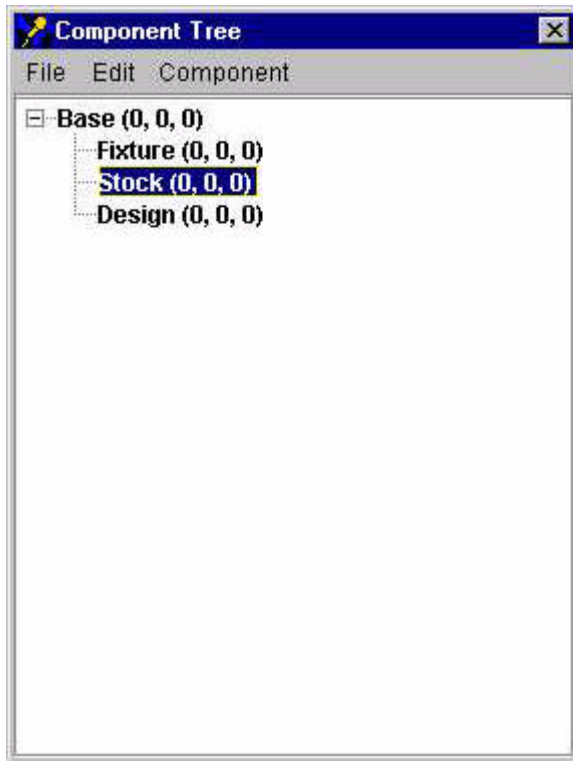
Setup > Control > Open
Shortcut=CGTECH_LIBRARY
File Name=yasmx3.ctl > Open

Define Components from "Base" to "Tool"

The components to be defined on the tool side of themachine are: Base > Z > Tool

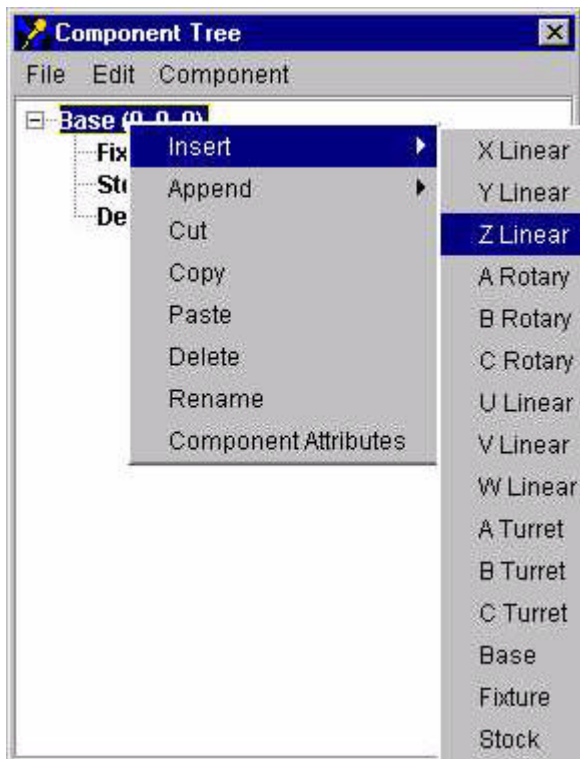
4. Display the Component Tree

Model > Component Tree



5. Add "Z" to "Base"

Select the **Base (0, 0, 0)** in the Component Tree
With the cursor on **Base**, Right click to pop up the menu
Insert > Z Linear



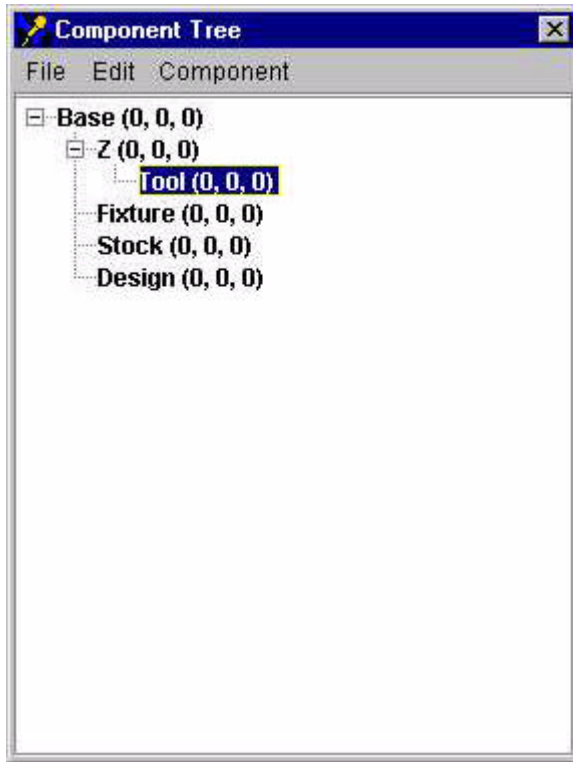
6. Add "Tool" to "Z"

Select the **Z (0, 0, 0)** in the Component Tree

With the cursor on **Z (0, 0, 0)**, Right click to pop up the menu

Append > Tool

Components Tree after adding "Tool side" components:



Define Components from "Base" to "Stock"

The components to be defined on the stock side of themachine are: **Base > Y > X > Stock**

7. Add "Y" to "Base"

Select the **Base (0, 0, 0)** in the Component Tree

With the cursor on **Base**, Right click to pop up the menu

Insert > Y Linear

8. Add "X" to "Y"

Select the **Y (0, 0, 0)** in the Component Tree

With the cursor on **Y (0, 0, 0)**, Right click to pop up the menu

Append > X Linear

9. Connect Design to stock

Select the **Design (0, 0, 0)** in the Component Tree
With the cursor on **Design (0, 0, 0)**, Right click to pop up the menu
Component Attribute
Connect To=Stock
Apply

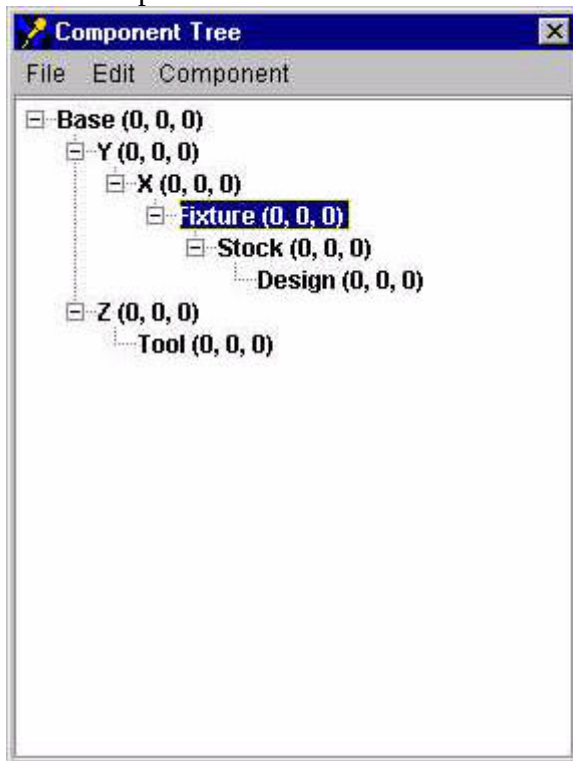
Components summary list after adding "Stock side" components:

10. Connect stock to Fixture

Select the **Stock (0, 0, 0)** in the Component Tree
In the **Component Attribute** Window:
Connect To=Fixture
Apply

11. Connect Fixture To X axis

Select the **Fixture (0, 0, 0)** in the Component Tree
With the cursor on **Fixture (0, 0, 0)**, Right click to pop up the menu
Cut
With the cursor on **X (0, 0, 0)**, Right click to pop up the menu
Paste
Expand the Tree under the **Fixture** and the **Stock** by pressing the +



12. Close the Component Attributes window and save a "3axmill.mch" Machine file

Component Attribute window: Close

Setup > Machine > Save As

Shortcut=Working Directory

File Name=3axmill.mch

Save

13. Set VERICUT to process G-Codes in a Tool Tip mode.

Setup > Toolpath > Toolpath Type=G-Code Data

Add

Shortcut=CGTECH_LIBRARY

File Name=gtooltip.mcd

OK > OK

Setup > G-Code > Settings > Settings Tab

Programming Method=Tool Tip

OK

14. Add a Stock model to be a 4.4 x 3.4 x 1.2 block

Model > Model Definition: Model tab

Type=Block

Length(X)=4.4

Width(Y)=3.4

Height(Z)=1.2

Add

Position Tab

Position=-0.2 -0.2 -1.2

Apply

OK

Fit 

15. Configure to retrieve tool data from the "gtooltip.tls" Tool Library file

Setup > Tool Manager

File > Open

Shortcut=CGTECH_LIBRARY

File Name=gtooltip.tls

Open

File > Close > Yes

16. Reset the model to ensure VERICUT is aware of changes to the machine and control, then open the Status window to monitor the simulation

Reset Model 

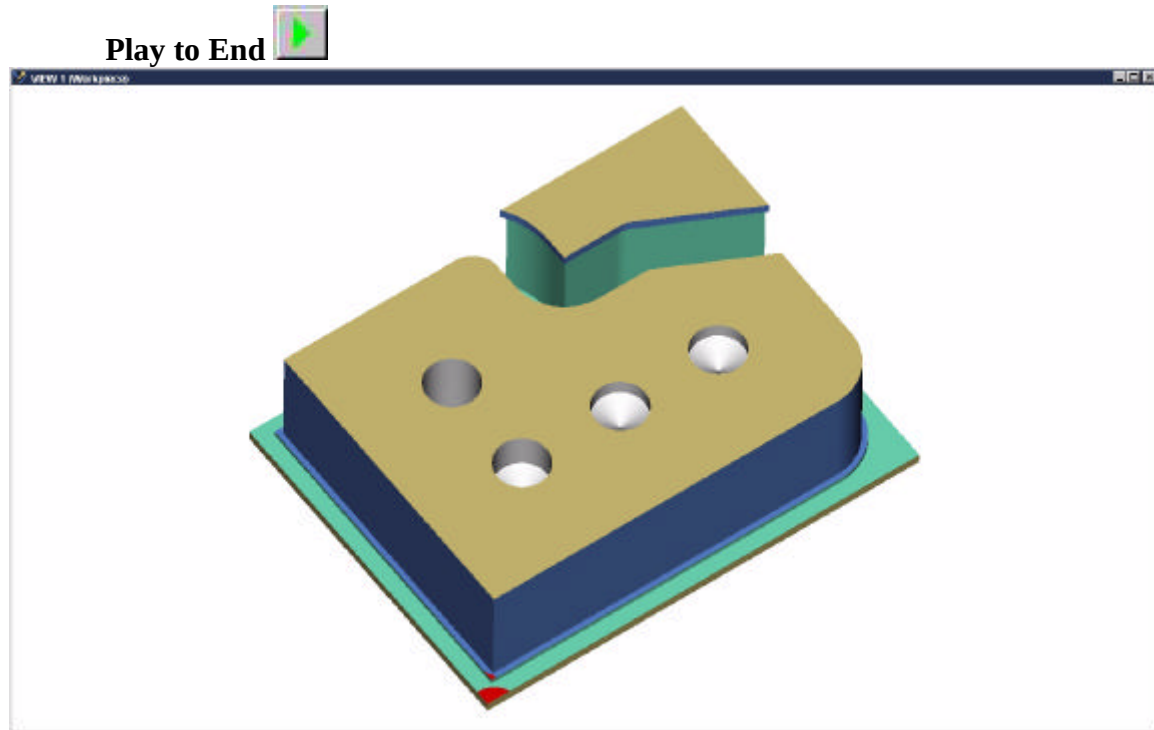
Info > Status...

Help on VERICUT

The status window is configured to show machine axis and tool tip locations. They should display the same values during the simulation.

Since VERICUT is now configured with the correct stock, tool path/orientation, and tools for simulating the "gtooltip.mcd" G-code tool path file, all that is left is to press "Play".

17. Cut the model

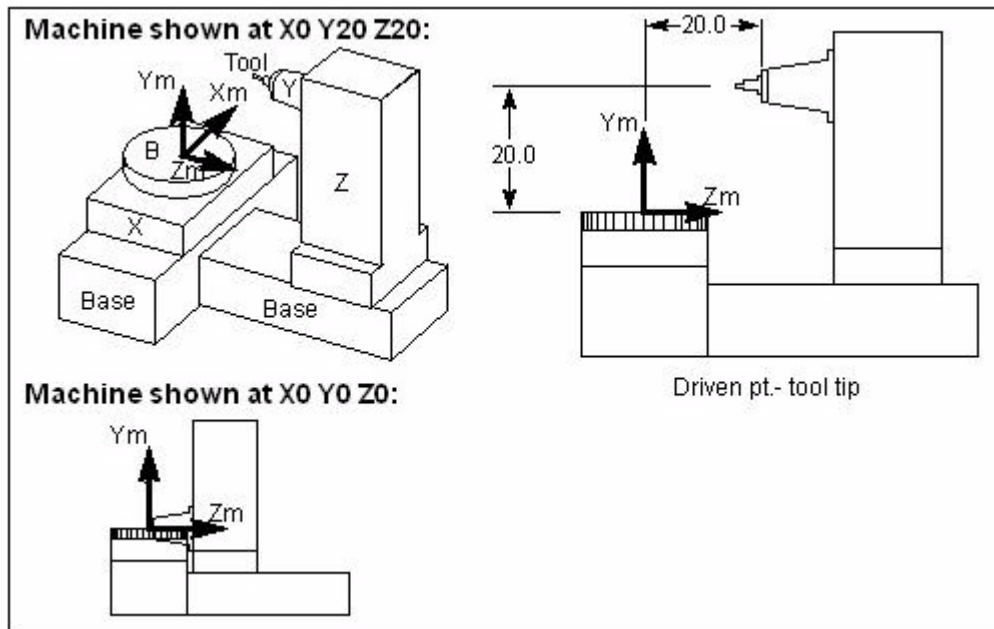


Session 23- Build a Kinematic Model of a 4-axis Horizontal Mill

This session shows how to configure a VERICUT Machinefile that describes the kinematics of a 4-axis horizontal mill with a "B" rotary table. A Cincinnati Acramatic 950 control interprets the G-code commands that are programmed to drive the tool tip, also known as "tool tip" programming method.

This machine can not be located at X0 Y0 Z0 B0 (machine zero) without causing a collision between the spindle and rotary table (see picture below). An Initial Machine Location table is used to move the machine to a safe starting location (X0 Y20 Z20) prior to cutting.

After configuring the job setup (machine, control, programming method, etc.) VERICUT is then configured to simulate the G-code tool path cutting action. Finally, a User file is saved enabling the simulation to be easily repeated by loading the User file and starting the simulation.



4-Axis Horizontal Mill

Session Steps:

1. Start from a new Inch User file

File > Properties

Default units=Inch > OK

File > New

2. Display the model, the Component and the Machine axis systems

View > Axes

Select **Model**

Select **Component**

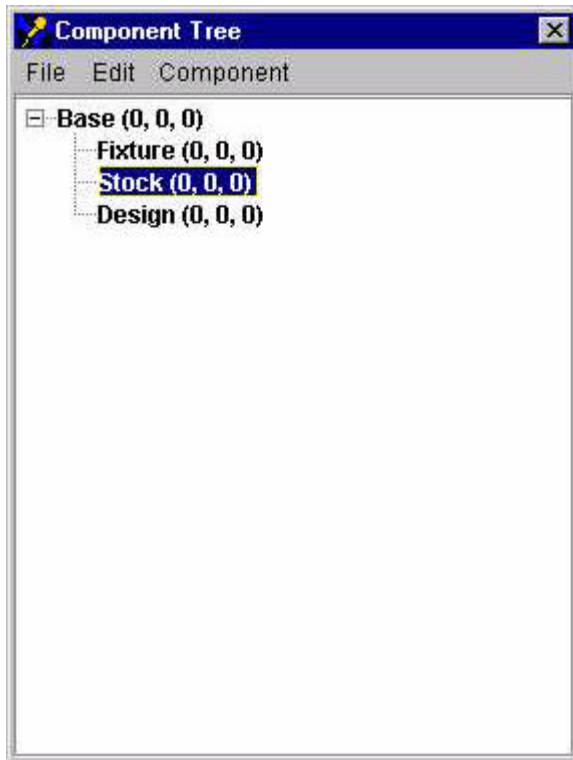
Select **Machine Origin**
Close

Define Components from "Base" to "Tool"

The components to be defined on the tool side of themachine are: **Base > Z > Tool**

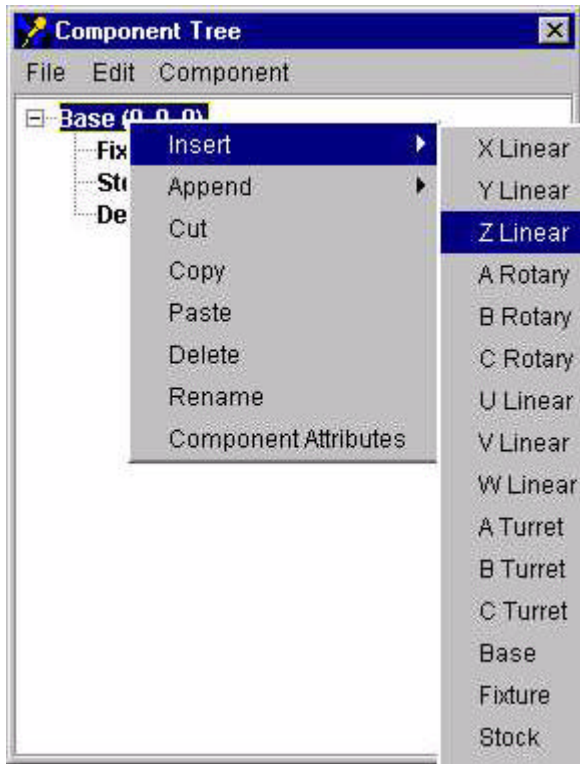
3. Display the Component Tree

Model > Component Tree



4. Add "Z" to "Base"

Select the **Base (0, 0, 0)** in the Component Tree
With the cursor on **Base**, Right click to pop up the menu
Insert > Z Linear



5. Add "Y" to "Z"

Select the **Z (0, 0, 0)** in the Component Tree

With the cursor on **Z (0, 0, 0)**, Right click to pop up the menu

Append > Y Linear

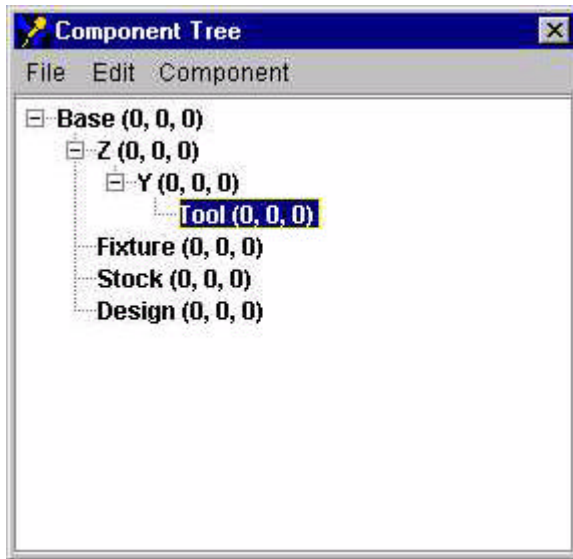
6. Add "Tool" to "Y"

Select the **Y(0, 0, 0)** in the Component Tree

With the cursor on **Y (0, 0, 0)**, Right click to pop up the menu

Append > Tool

Components Tree after adding "Tool side" components:



Define Components from "Base" to "Stock"

The components to be defined on the stock side of themachine are: Base > X > B > Stock

7. Add "X" to "Base"

Select the **Base (0, 0, 0)** in the Component Tree
With the cursor on **Base**, Right click to pop up the menu
Insert > X Linear

8. Add "B" to "X"

Select the **X (0, 0, 0)** in the Component Tree
With the cursor on **X (0, 0, 0)**, Right click to pop up the menu
Append > B Rotary

9. Delete Design and Fixture

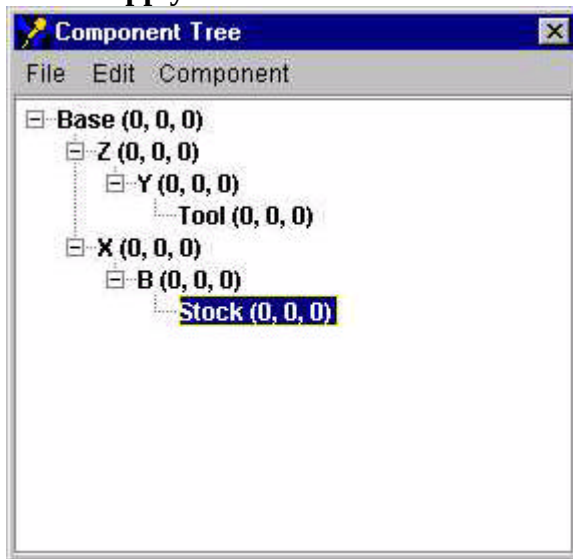
Select the **Design (0, 0, 0)** in the Component Tree
With the cursor on **Design (0, 0, 0)**, Right click to pop up the menu
Delete > Yes
Select the **Fixture (0, 0, 0)** in the Component Tree
With the cursor on **Fixture (0, 0, 0)**, Right click to pop up themenu
Delete > Yes

Components summary list after adding "Stock side" components:

10. Connect stock to "B"

Select the **Stock (0, 0, 0)** in the Component Tree
With the cursor on **Stock (0, 0, 0)**, Right click to pop up the menu
Component Attributes
In the **Component Attributes** Window:
Connect To=B

Apply



11. Define an "Initial Machine Location" table to locate the machine at X0 Y20 Z20

Setup > Machine > Settings
Table Tab > Table Name=Initial Machine Location
Values (XYZABCUVW)=0 20 20
Add > OK

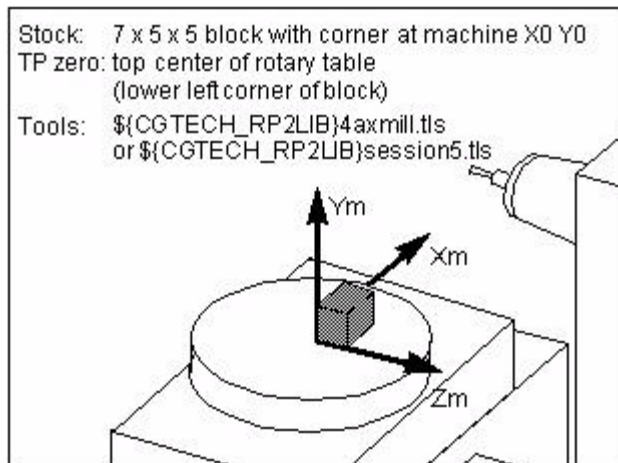
12. Close the Component Attributes window and save a "4axmill.mch" Machine file

Component Attribute window: OK
Setup > Machine > Save As
Shortcut: Working Directory
File Name=4axmill.mch
Save

13. Open a different user file "vcblock2 usr"

File > Open > No
Shortcut=CGTECH_SAMPLES
File Name=vcblock2.usr
Open

See the job setup diagram below for stock size and tool path location for proper machining.



14. Set VERICUT to process G-Codes in a Tool Tip mode.

Setup > Toolpath > Toolpath Type=G-Code Data
Select vcblock.tp > Replace
Shortcut=CGTECH_SAMPLES
File Name=4axmill.mcd
OK > OK
Setup > G-Code > Settings > Settings Tab
Programming Method=Tool Tip
OK

15. Add a Stock model to be a 7 x 5 x 5 block

Model > Model Definition: Model tab
Stock size ok as is
OK

16. Configure to retrieve tool data from the "4axmill.tls" Tool Library file

Setup > Tool Manager
File > Open
Shortcut=CGTECH_SAMPLES
File Name=4axmill.tls
Open
File > Close > Yes

17. Open the Acramatic 950MC milling Control File

Setup > Control
File > Open
Shortcut=CGTECH_LIBRARY
File Name=acr950mc.ctl
Open

18. Open the new 4 axis vertical mill file "4axmill.mch"

Setup > Machine
File > Open
Shortcut=Working Directory
File Name=4axmill.mch
Open

17. Reset the model to ensure VERICUT is aware of changes to the machine and control, then open the Status window to monitor the simulation

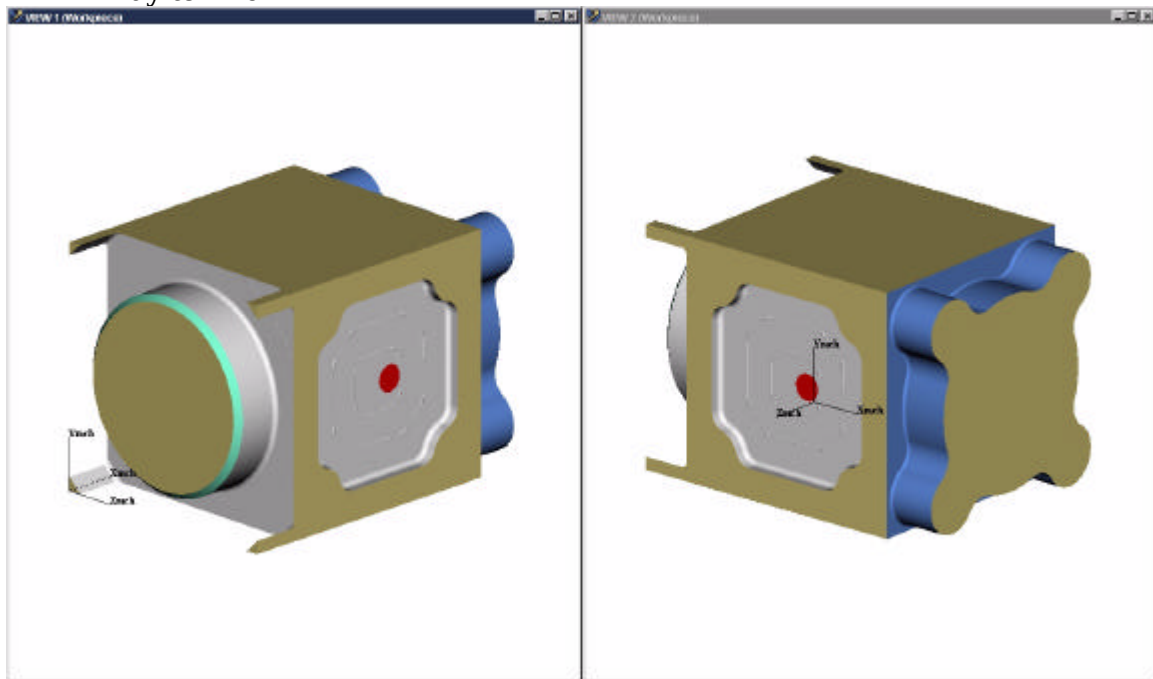
Reset Model 
Info > Status

The status window is configured to show machine axis and tool tip locations. They should display the same values during the simulation.

Since VERICUT is now configured with the correct stock, tool path/orientation, and tools for simulating the "gtooltip.mcd" G-code tool path file, all that is left is to press "Play".

17. Cut the model

Play to End 



Chapter 6 **ERRORS and Problem Solving**

Error messages

This section describes error messages and problems which may occur while operating VERICUT. Look for error messages in the VERICUT main window message area, or in the Log file created during the simulation.

Note: Licensing and license server-related errors that occur when attempting to run VERICUT are described in the "ERRORS / Problem Solving" section in the *VERICUT Installation* documentation.

Common error messages are listed below in alphabetical order (case and special characters not considered), along with problem descriptions and recommended solutions.

See also: "Error conditions" in this chapter.

Bad cutter description - height not specified

Problem: cutter description does not have a specified height.

Solution: correct the cutter description to specify a positive height, or specify a default height via **Setup menu > Motion: Min. Cutter Height**.

Cannot find cutter data for [n] in tool library file...

Problem: tool change record processed requesting tool data for tool "n", but Tool Library does not contain tool data for ID "n".

Solution: use the Tool Manager to add a tool under ID "n" to the Tool Library, then reset and re-process the tool path file.

Cannot locate sequence number 0 for GOTO

Problem: attempted to process an APT tool path containing a "GOTO" record when **Toolpath Type=G-Code Data**.

Solution: enter the appropriate G-code data tool path file name, or choose another **Toolpath Type** option to process this tool path file.

Cannot locate sequence number n for GOTO

Problem: processed a branch command to sequence number "n", but this sequence number does not exist in the G-code tool path file or subroutines.

Solution: correct the G-code tool path or subroutine files to have a sequence number "n", or do not reference this sequence number in the branch command.

Cannot process TLDATA type...

Help on VERICUT

Problem: TLDATA record processed containing GROOVE or THREAD data which is not interpreted.

Solution: use the Tool Manager to add a tool to a Tool Library, create a tool list that references the tool in the library, then reset and re-process the tool path file.

Collision with clamp or fixture occurred at record sequence number...

Problem: a tool has removed material from a Fixture model. VERICUT assumes that fixtures are not to be machined. The cut in the fixture is shaded using the **Error** color, typically red.

Solution: if the fixture was not intended to be cut, use the tool path history information associated with the error in the Log file to identify the error tool path record for correction.

Collision with design model occurred at record sequence number...

Problem: a tool has removed material from a Design Solid model when the **Constant Gouge Check** is on. The cut in the design model is shaded using the **Error** color, typically red.

Solution: use the tool path history information associated with the error in the Log file to identify the error tool path record for correction.

Collision with protected stock occurred at record sequence number...

Problem: a tool has removed material from a previously machined surface protected by the **Protect Cuts** feature. The cut in the previously machined surface is shaded using the **Error** color, typically red.

Solution: use the tool path history information associated with the error in the Log file to identify the error tool path record for correction.

Could not find OpenGL Visual...

Problem: Pre-5.0 version of VERICUT executed with the “opengl” command line option on a computer that does not have Open/GL display capability, or computer that does not yet have support for Open/GL display. When this error occurs, VERICUT uses the standard “X” display mode instead.

Solution: ensure that the computer is supported by CGTech for Open/GL display capability. Consult the CD booklet or *VERICUT Installation* documentation for details about computers on which VERICUT can be operated in Open/GL display mode. If supported, ensure that the Open/GL hardware and software are installed and operational.

Cut depth ... exceeds maximum OptiPath table depth...

Problem: cutting tool being optimized by **Feed/Depth Tables** has cut deeper than the deepest specified cut depth.

Solution: use as is or adjust the **Depth** table values, then reset and re-optimize the tool path file.

Cut width ... exceeds maximum OptiPath table width...

Problem: cutting tool being optimized by **Feed/Depth Tables** has cut wider than the widest specified cut width.

Solution: use as is or adjust the **Width** table values, then reset and re-optimize the tool path file.

Error for line 0 - cannot open Job file

Problem: the Job file does not exist or have read permissions.

Solution: ensure that the specified Job file exists. If it exists, verify that the file can be read by the current user.

Error for line 0 - reading job file...

See "Error for line 0 - cannot open Job file".

Error for line 0 - unable to check out feature...

Problem: the license server is not able to grant the request for the CGTech product/option. This general error message is followed by additional error messages which more accurately describe the problem.

Consult the secondary error message in this section for the problem/solution related to the above error message.

Error for line 0 ... (n) - Error while processing apt cutter

Problem: Tool Library file contains an invalid cutter description at line number "n".

Solution: edit the cutter description to be valid, then re-load the Tool Library file.

Error for line 0 ... (n) - Error while processing primitive

Problem: Tool Library file contains an invalid tool primitive at line number "n".

Solution: edit the tool primitive to be valid, then re-load the Tool Library file.

Error for line 0 ... (1) - expecting file header

Problem: the identified VERICUT file does not have a valid file header.

Solution: consult "File Types and Descriptions" in the *VERICUT Files* chapter for information about valid VERICUT file formats.

Problem: V1.4 Tool Library file has been referenced from a V3.x VERICUT system.

Solution: V1.4 Tool Library files can not be loaded in V3.x systems. Use the Tool Manager to create a new Tool Library file.

Problem: errors prevent a Tool Library file from being successfully loaded, however, not enough information is present to determine the source of the error(s).

Solution: re-attempt to load the Tool Library file and check the message areas and Log file for additional error messages which describe the error condition.

Error for line 0 ... (n) - syntax error

Problem: the identified Tool Library file contains an invalid tool description at line number "n".

Help on VERICUT

Solution: edit the record to be valid (or delete the record if not needed). After editing, re-load the Tool Library file.

Error for line n

Problem: an error has been detected by VERICUT while processing line number “n” in the tool path file.

Solution: check the message areas and Log file for additional error messages which describe the error condition.

Error loading tool library file...

Problem: the identified Tool Library file contains an invalid tool description.

Solution: consult the Log file to identify the first error record in the Tool Library file, then edit the record to be valid (or delete the record if not needed). After editing, re-load the Tool Library file.

Fast feed rate removed material...

Problem: A tool has removed material with an unsafe feed rate. This error also commonly occurs when a metric unit tool path is processed after loading a User file configured to process an inch unit tool path file. The cut color of this error motion is shaded using the **Error** color, typically red.

Solutions:

The tool has unexpectedly removed material with an unsafe feed rate- use the tool path history information associated with the error in the Log file to identify the responsible tool path record.

Tool motion is okay, but the Fast Feed value is set too low for the current tool- open **Setup menu > Motion** and set the **Fast Feed** value as needed for the tool. If safe cutting feed rates vary per tool, consider adjusting this value frequently, or inserting VERICUT comments records to automatically adjust it. For more information, see "PPRINT/VERICUT-MODAL record" in the *VERICUT Files* chapter.

HOLDER removed material at record sequence number...

Problem: a tool holder has removed material. The error cut is shaded using the **Error** color, typically red.

Solution: use the tool path history information associated with the error in the Log file to identify the error tool path record for correction.

Invalid circle... r1= ... , r2= ...

“r1” is the calculated distance from the circle center to the circle start point. “r2” is the calculated distance from the circle center to the circle end point.

Problem: *Small differences between r1 and r2 values-* the current **Circle Radial Tolerance** value differs from what the actual NC control uses.

Solution: Access **File menu > Properties: Tolerance** tab and increase the tolerance amount to allow for the calculated differences between r1 and r2.

Help on VERICUT

Problem: *Large differences between r1 and r2 values-* control settings for are not correctly configured to interpret circle motion data.

Solution: Access **Setup menu > Control > Settings > Circles** and configure the window as required to reflect how the IJK values in the G-code tool path file should be interpreted.

If the above solutions do not suppress correct the problem, then circle motion data in the tool path is incorrect. Correct the programmed motion and re-process the tool path file.

Invalid CYCLE - does not match any of the definitions...

Problem: A CYCLE record is not compatible with any of the current cycle definitions.

Solution: Use the features on **Setup menu > APT Settings: Cycles** tab to configure VERICUT to understand your cycle commands.

If the above solutions do not suppress correct the problem, then cycle motion data in the tool path is incorrect. Correct the programmed motion and re-process the tool path file.

Licensed number of users already reached (-4,132) (-4)

Problem: all available licenses for a requested CGTech product/option have been checked out.

Solution: contact your CGTech representative for information about obtaining additional CGTech product licenses.

Machining type mismatch for INSERT Tool not defined - motion ignored

Problem: V1.5-2.x Tool Library file containing invalid turning cutter descriptions has been loaded. All cutter descriptions must have the text "TURN," at the start of each description.

Solution: edit turning cutter descriptions in the Tool Library file to have the text "TURN," at the start of each description. For example, edit "1:.060,,,,55,15,.5" to be "1:TURN,.060,,,,55,15,.5". Then, re-load the Tool Library file.

Maximum number of errors has been reached

Problem: the maximum number of errors specified via **Setup menu > Motion: Stop At Max Errors** has been detected.

Solution: restart cutting to continue, or enter a larger number of errors to be detected before processing stops.

See also other error messages following this message.

Modified toolpath directory is not write accessible

Problem: V3.3 VERICUT erroneously outputs this message when attempting to optimize multiple tool paths in a tool path file list.

Solution: obtain and use a version V3.3.1 or higher of VERICUT.

modified (optimized) tool path file can not be accessed or opened by the current user

Problem: file permissions prevent the optimized tool path file from being written.

Help on VERICUT

Solution: change file access permissions to allow user access. For example, UNIX use the “chmod” command: “chmod 777 example” enables the file “example” to be executed, written to, or read by all users.

No results file defined

Problem: tried to view a Results file that has not been created.

Solutions: prior to processing, enter a name in the **Results file** field on the **File menu** >

Properties: Results tab.

Process fork failed

Problem: the computer’s local memory has been exhausted.

Solution: exit all unused applications and close unused windows to free system resources, then retry. If the condition persists exit from all CGTech products and reboot the computer.

SET ACCESS NOT ALLOWED

Problem: the directory from which CATV has been executed does not have “write” permissions for the current user.

Solution: exit the CATIA/CATV session, change directories to one that files can be written to, then re-execute CATIA/CATV.

SHANK removed material at record sequence number...

Problem: a tool shank has removed material. The error cut is shaded using the **Error** color, typically red.

Solution: use the tool path history information associated with the error in the Log file to identify the error tool path record for correction.

The Word ... is not defined

Problem: the G-code word “xxx” is not defined in the NC control configuration.

Solution: access **Setup menu** > **Control** > **Word Format** and configure the control to process this word.

Problem: attempted to process an APT or tool path type other than G-code data when the **Toolpath Type=G-Code Data**.

Solution: enter the appropriate tool path file name, or choose another **Toolpath Type** option to process this tool path file.

Tool Library file data is only partially loaded

Problem: Tool Library file contains an invalid cutter description.

Solution: see the Log file for additional error messages which describe the error condition. Edit the cutter description to be valid, then re-load the Tool Library file.

Tool not defined - motion ignored

Help on VERICUT

If this error is preceded by the error “Machining type mismatch for INSERT”, see "Machining type mismatch for INSERT Tool not defined - motion ignored".

Problem: a valid tool description has not been supplied to VERICUT at the time tool path processing was started.

Solution: access **Setup menu** > **Toolpath** to indicate where cutting tool descriptions can be found.

Error conditions

This section describes error conditions which may occur while operating VERICUT, along with problem descriptions and recommended solutions. These conditions do not have related error messages. To determine problems and solutions related to error messages, consult "Error messages" in this chapter.

Circle motions are not displayed as expected

This error condition is typically flagged by an error message similar to "Invalid circle... r1= , r2=" where "r1" is the calculated distance from the circle center to the circle start point. "r2" is the calculated distance from the circle center to the circle end point.

Problem: *Small differences between r1 and r2 values*- the current **Circle Radial Tolerance** value differs from what the actual NC control uses.

Solution: Access **File menu > Properties: Tolerance** tab and increase the tolerance amount to allow for the calculated differences between r1 and r2.

Problem: *Large differences between r1 and r2 values*- control settings for are not correctly configured to interpret circle motion data.

Solution: Access **Setup menu > Control > Settings > Circles** and configure the window as required to reflect how the IJK values in the G-code tool path file should be interpreted.

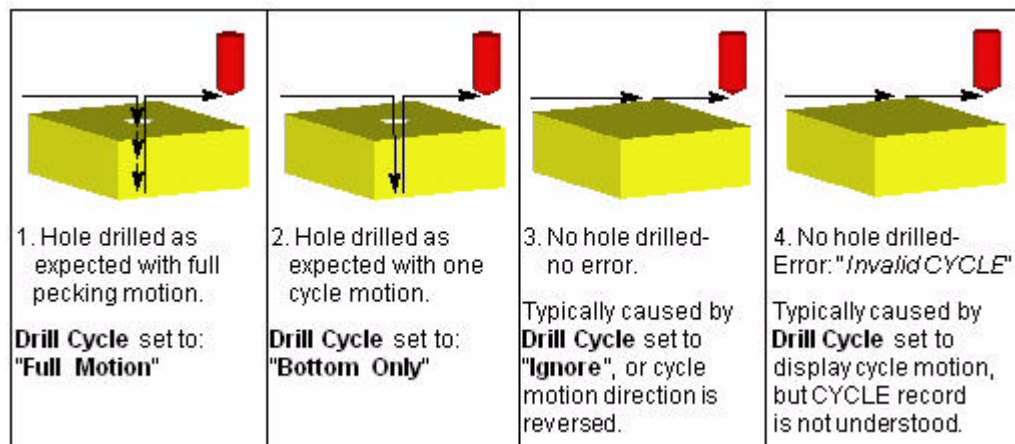
Problem: If the above solutions does not suppress the message, then circle motion data in the tool path is not correct.

Solution: Correct the circle motion G-code data and re-process the tool path file.

If the above solutions do not suppress correct the problem, then circle motion data in the tool path is incorrect. Correct the programmed motion and re-process the tool path file.

Cycle motions are not displayed as expected

Consult the figure below to troubleshoot incorrect cycle motion scenarios.



Help on VERICUT

Scenario #1-2 - No problems.

Scenario #3

Problem: APT-CLS or G-code cycle motion is not being simulated-cycles are being ignored.

Solution: Click **Setup menu > Motion** and set **Drill Cycle** to something other than "**Ignore**".

Problem: Cycle is simulated, but motion direction is away from the workpiece. You may have to look closely, or slow **Animation Speed** to see this.

Solution: Use features on **Setup menu > APT Settings: Cycles** tab to edit cycle definitions to be correct for your cycle commands.

Scenario #4

Problem: APT-CLS CYCLE record is not compatible with any of the current cycle definitions.

Solution: Use features on **Setup menu > APT Settings: Cycles** tab to configure VERICUT to understand your cycle commands.

If the above solutions do not suppress correct the problem, then cycle motion data in the tool path is incorrect. Correct the programmed motion and re-process the tool path file.

Input File window highlights all records (turns black) when the tool path is processed

Problem: A "Begin Data" word is defined in the control configuration, but was not encountered in the tool path file.

Solution: Access **Setup menu > Control > Word Format** and delete the word definition which designates the beginning of data, typically "%".

Index

- # of Cuts - AutoSave window, 270
- # Teeth list - OptiPath Manager window, 402
- #Teeth - Tool Properties window, 344
- % Feedrate/Angle table - Adv. OP Settings window
 - Angle tab, 402
- % Width/% Feedrate table - OptiPath Manager window
 - Width Table tab, 402
- 10 Parameter cutter - Tool Manager mill tool, 338
- 4 Parameter cutter - Tool Manager turn tool, 338
- 4.4 functions - map of locations in 5.0, 222
- 4-Axis Sync Method - G-Code Settings window
 - Turning tab, 318
- 4-axis turning, 318
 - G-code settings for, 318
 - see also "Turning", 318
- 7 Parameter cutter - Tool Manager mill tool, 338
- A, 508
- A Rotary - component type, 307
- A Turret - component type, 307
- A-axis Rotary Type - Rotary control settings window, 369
 - CUTTER record parameter, 508
 - Tool Manager mill tool parameter, 338
 - Tool Manager turn tool parameter, 338
- About building controls, 197
- About building machines, 190
- About CGTech documentation, 16
- About configuring for cycle simulation, 138
- About configuring tables, 191
- About modeling, 85
- About optimizing tool paths, 221
- About simulating subroutines, 208
- Absolute - Tool Manager "Position" window, 347
- Absolute Rotary Direction - Rotary control settings window, 369
- Accel/Decel - Modeling window
 - Component Attributes tab, 304
- Accel/Decel - OptiPath Manager window, 402
- Acceleration - Modeling window
 - Component Attributes tab, 304
- Accessing VERICUT, 14
- ACL - tool path type, 311
- Acramatic (Cincinnati) control, 528
 - configuring cycles for, 366
 - library control files, 528
- Active Component - Modeling window, 297
- active view, 275
- Add, 67
- Add-user file, 481
 - Add More Cuts - OptiPath Manager window
 - Settings tab, 402
 - collision detect case, 387
 - component, 88
 - coordinate system, 130
 - cut color, 79
 - description of, 481
 - display color, 77
 - event-driven macro or variable, 204
 - model, 97
 - open via command line option, 467
 - optimization record to OptiPath Library, 402
 - subroutine file name, 149
 - substitution - to control config., 206
 - table or table row, 147
 - tool change event to a tool list, 128
 - tool path to simulate, 116
 - tool to Tool Library, 170
 - variable, 145
 - view, 67
 - view to view list, 73
 - word or special character, 200
 - Word/Address class, 201
 - Word/Address group, 202
- addusr= - command line option, 467
- Adjust Feedrate for Entry Angle - OptiPath Manager window
 - Angle tab, 402
- Adv. Options - Setup menu > Control, 381
- Air Cut Feed Rate - optimization method and value, 402
- Air Distance, 391
 - measuring, 184
 - X-Caliper window tab, 391
- Allen-Bradley control, 528
- Analysis - Toolbar group, 285
- Analysis menu - VERICUT main window, 229
- Analyze - see "Measure", 391, 394
- Angle - OptiPath Manager window tab, 402
- Angle Value - APT Settings window
 - Rotary tab, 331
- Angle/% Feedrate table - OptiPath Manager window
 - Angle tab, 402
- Angles, 299
 - Angles and coordinate systems, 16
 - Coordinate System window, 336

Help on VERICUT

- measuring between features, 180
- Modeling window
 - Position tab, 299
- Tool Manager "Position" window, 347
- View Orient window, 277
- Animation, 265
 - Animation Control - Image Record window, 265
 - Animation Speed - Motion window, 331
 - controlling speed of, 175
 - Record Images window, 265
 - speed of Image playing back, 266
 - speed of VERICUT simulation, 315
- Annotation - ShadeCopy window
 - Postscript Settings tab, 268
- Append, 265
 - Append Delay - Image Record window, 265
 - appending a component, 307
 - to Image file - interactively, 265
 - to Image file - via command line option, 467
 - to Log file - via command line option, 467
- append_log - command line option, 467
- Apply Switch Value - G-Code Settings window
 - Block Skip tab, 318
- Apply Tool Nose Comp with CDC - Cutter
 - Compensation control settings window, 375
- APT binary data - convert to ASCII APT tool path, 111
- APT Output file, 481
 - creating via reverse post-processing, 154
 - description of, 481
 - specifying the file name, 324
- APT Output for Circular Motion - Circles
 - control settings window, 365
- APT Settings - Setup menu, 331
- APT Table file, 481
 - Binary CL converter window, 249
 - description of, 481
- APT tool path file, 481
 - creating from binary APT data, 111
 - creating from NCI data, 112
 - description of, 481
 - description of records in, 508
- APT tool path records, 508
- Arc - Tool Manager mill tool profile, 338
- Arc - Tool Manager turn tool profile, 338
- Assemble, 301
 - component or model, 95
 - Modeling window tab, 301
 - tool holder, 347
 - Tool Manager "Position" window, 347
- Assign - Color window tab, 257
- Attributes, 279
 - components, 304
 - ShadeCopy, 268
 - Tool Manager > View menu function, 338
 - View menu function, 279
- Auto - Section window, 281
- Auto Block - Model window
 - Model tab, 298
- Auto Error - AutoSave window, 270
- Auto Save, 270
 - IP files, 270
 - options on AutoSave window, 270
 - ShadeCopy images, 270
- Auto Sequence Numbering - OptiPath control settings window, 378
- auto_mode - cycle def. modal, 138
- autodemo= - command line option, 467
- AUTO-DIFF, 19
 - Analysis menu function, 394, 508
 - AUTO-DIFF Report file, 481
 - command line option, 467
 - license description, 19
 - PPRINT control of, 508
- Automatic sectioning, 281
- AutoSave - File menu function, 270
- Axes - Tool Manager-View Attributes window, 338
- Axes - View menu, 284
- Axial Depth - OptiPath Manager window
 - Feed/Speed tab, 402
- Axis - X-Caliper window
 - Scallop tab, 391
- Axis Priority - Machine Settings window, 387
- Axis Priority list - Mach. Settings window
 - Axis Priority tab, 387
- B, 508
- B Rotary - component type, 307
- B Turret - component type, 307
- Background, 257
 - color definition, 257
 - types, 279
 - view attribute, 279
- Backward - Tool Manager window, 338
- Ball center programming - setting the control point for, 315
- Ball Nose End Mill - Tool Manager mill tool, 338
- Base - component type, 307
- Base Location, 245
 - Modeling window
 - Import tab, 306
 - Surface to Solid window, 245
- Base Radius - Modeling window
 - Model tab, 298
- Base Work Offset table, 419
- batch - command line option, 467
- Batch processing, 467, 478
 - command line options for, 467
- batchp - command line option, 467

Help on VERICUT

- batcht - command line option, 467
- B-axis Rotary Type - Rotary control settings window, 369
 - CUTTER record parameter, 508
 - Tool Manager mill tool parameter, 338
 - Tool Manager turn tool parameter, 338
- Binary APT data - convert to ASCII APT tool path, 111
- Binary CL converter, 249, 481
 - APT Table file used with, 481
 - command line options, 474
 - converting binary data, 111
 - description of, 249
 - window features, 249
- Block, 298
 - Block Skip - G-Code Settings window tab, 318
 - Block Skip Character - G-Code Settings window
 - Block Skip tab, 318
 - block skipping - how to configure, 152
 - model type, 298
 - Tool Manager turn tool holder, 338
- Blocking factor, 481
- Bosch control, 528
- Bottom - ShadeCopy window
 - Postscript Settings tab, 268
- Bottom Radius - Tool Manager turn tool, 338
- Boundary - AUTO-DIFF window, 394
- Branching logic - see "Looping-branching logic", 327
- Brightness - Color window
 - Define tab, 257
- Build Tool List - Tool Change List window, 311
- Building controls, 197
- Building machines, 190
- Bull Nose End Mill - Tool Manager mill tool, 338
- BUSY light, 41
- Button - Tool Manager turn tool, 338
- C Rotary - component type, 307
- C Turret - component type, 307
- Caclulate minimum cutter extension, 137
- Caclulate minimum cutter height, 137
- Calculate Min. Cutter Extension - Motion window, 315
- Calculation Tolerance - General control settings window, 362
- Canned cycles - see "Cycles", 366
- Cascade - views, 68
- Catia Interface - license, 19
- CATIA0 record, 508
- CATV - license, 19
- C-axis Rotary Type - Rotary control settings window, 369
- CCW Motion Cancels Cycles - Motion control settings window, 363
- CDC - see "Cutter compensation", 327
- CDC ON/OFF Method - Cutter Compensation control settings window, 375
- CDC Ramp ON/OFF with - Cutter Compensation control settings window, 375
- Ceiling, 279
 - background attribute, 279
 - color assignment, 257
 - displaying, 72
 - setting "Machine Up" direction for, 387
- Center, 268
 - Center - X-Caliper window
 - Scallop tab, 391
 - Center of Rotation - Coordinate System window, 336
 - Center of Rotation - Modeling window
 - Position tab, 299
 - Center of Rotation - Tool Manager "Position" window, 347
 - Postscript image for printing, 268
- CGTech - contact info, 1
- CGTech Macro Executable file - description of, 481
- CGTECH_DEFAULT_UNITS - env. variable, 479
- CGTECH_FONT_SIZE - env. variable, 479
- CGTECH_LIBRARY - env. variable, 42
- CGTECH_STATUS - env. variable, 479
- CGTECH_USERNAME - env. variable, 479
- CGTECH_VC_IP_FILTER - env. variable, 479
- CGTECH_VC_USR_FILTER - env. variable, 479
- CGTECH_VCUSR - env. variable, 479
- Change Record - Status window, 289
- Change tools, 119
 - manual tool change, 122
 - overriding current tool, 124
 - setting an initial tool, 123
 - using a tool list, 126
 - via cutter description records, 119
 - via program stop, 122
 - via tool path file names, 121
 - via tool/pocket number records, 120
- change units - for a new VERICUT session, 44
- Charmilles control, 528
- Check Limits - AUTO-DIFF window
 - Tolerance tab, 394
- Chip Thickness - optimization method, 402
- Cincinatti control - see "Acramatic", 528
- Circle, 508
 - Circle Center Mode - Circles control settings window, 365
 - Circle Feedrate - OptiPath Manager window
 - Settings tab, 402

Help on VERICUT

- Circle Radial Tolerance - Properties window
 - Tolerance tab, 261
- CIRCLE record, 508
- Circle Tolerance - Mastercam converter, 250
- Circles, 508
 - APT-CLS record formats, 508
 - configuring control settings for, 210
 - error messages and conditions, 674, 681
 - Setup > Control > Settings function, 365
 - tolerance used for circle radial calculations, 261
- Class list - Word/Address window, 357
- Clean-up Feedrate - OptiPath Manager window
 - Settings tab, 402
- Clear, 240
 - Adv. Control Options window
 - Subroutines tab, 381
 - Binary CL converter message area, 249
 - Clear All - View Axes window, 284
 - Clear All - View Toolbar window, 285
 - G-Code Settings window
 - Subroutines tab, 318
 - IGES converter message area, 240
 - Mastercam converter message area, 250
 - OptiPath tool properties - Tool Properties window, 344
 - PolyFix converter message area, 243
 - subroutine file names, 149
 - Surface to Solid converter message area, 245
 - tool change events from a tool list, 128
 - Toolpath window, 311
 - VDA converter message area, 247
- Clearance Distance - OptiPath Manager window
 - Entry/Exit tab, 402
 - Plunge tab, 402
- CLEARP record, 508
- Close, 274
 - Component Tree window, 307
 - Toolbar, 285
 - VERICUT window (exit), 274
- Closest Point - X-Caliper window, 391
- CLRSRF record, 508
- CLS tool path file - description of, 481
- CLS tool path records, 508
- CME file, 481
 - Adv. control option, 381
 - description of, 481
 - specifying a CME file to use, 381
- CNTRL record, 508
- Collision Detect - Machine Settings window, 387
- Collision Detection On - Mach. Settings window
 - Collision Detect tab, 387
- Color, 76
 - assigning to cut sequences, 79
 - assigning to the cutting tool, 78
 - background - flat, 82, 257
 - background - shaded, 257
 - brightness, 83
 - ceiling color, 257
 - changing during a cut, 80
 - Color / Cut Seq. list - Color window
 - Assign tab, 257
 - Color list - AUTO-DIFF window
 - Tolerance tab, 394
 - Color List - Color window
 - Define tab, 257
 - component color, 304
 - controlling in VERICUT, 76
 - defining for display, 77
 - error color, 84
 - flat colors, 257
 - floor color, 257
 - foreground - flat, 82
 - model color, 298
 - resetting cut colors, 81
 - shade colors, 257
 - tool property, 344
 - wall color, 257
- Colors - File menu function, 257
- Command - OptiPath Data Link window, 416
- Command files, 479
 - bincl2v, 249
 - iges2v, 240
 - mcam2v, 250
 - polyfix, 243
 - prshade, 270
 - sur2stk, 245
 - vda, 247
 - vericut, 14
- Command line options, 467
 - converters - list of, 474
 - list of for use in VERICUT, 467
 - obsolete command line options, 467
- Comment records - see "PPRINT", 508
- Compare - AUTO-DIFF window, 394
- Comparison Method - AUTO-DIFF window
 - Settings tab, 394
- Comparison Type - AUTO-DIFF window
 - Settings tab, 394
- Component - View Axes window, 284
- Component 1-2 - Mach. Settings window
 - Collision Detect tab, 387
- Component Attributes, 304
 - about modeling and component attributes, 85
 - affect in modeling, 85
 - Component Tree > Edit menu function, 307
 - menu on Component Tree window, 307
 - specifying, 90
 - tab on Modeling window, 304
- Component menu - Component Tree window, 307

Help on VERICUT

- Component Min/Max travel list - Mach. Settings window
 - Travel Limits tab, 387
- Component Tree, 307
- Component Tree - about defining, 85
 - changing connect order, 91
 - from File > New, 232
 - Model menu function, 307
- Component Type - Component Tree > Edit menu, 307
- Component Type - Modeling window, 297
- Component/Component collision list - Mach. Settings window
 - Collision Detect tab, 387
- Components, 85
 - about defining, 85
 - adding, 88
 - assembling, 95
 - changing connect order, 91
 - changing types, 94
 - component tree, 307
 - copying, 88
 - deleting, 88
 - detecting collisions with, 193
 - displaying axes of, 284
 - moving, 95
 - positioning, 95
 - renaming, 93
 - rotating, 95
 - specifying attributes, 90
- Composite Format - Add/Modify Word window, 352
- Compute - X-Caliper window, 391
- Condition - Add/Modify Word/Address windows, 357
- Conditional functions - see "Functions", 463
- Conditional State/Value - Add/Modify Word/Address windows, 357
- Conditional Variable/Range - Add/Modify Word/Address windows, 357
- Conditional Word/Range - Add/Modify Word/Address windows, 357
- Cone, 298
 - model type, 298
 - Tool Manager turn tool holder, 338
- Configure - Image Playback window, 266
- Configure - Status window, 289
- Connect angle - Export Model window, 252
- Connect To - Modeling window
 - Component Attributes tab, 304
- constant chip thickness - optimization method, 402
- constant surface speed, 318
- constant volume - optimization method, 402
- Constraints, 301
 - components and models, 301
 - tool holders, 347
 - using during assembly, 95
- CONT record, 508
- Control file, 481
 - about building, 197
 - description of, 481
 - library Control files, 528
 - opening, 49
 - saving, 49
 - using MDI to test, 196
- Control Point, 315
 - Control Point Offset (X Y Z) - Tool Properties window, 344
 - Control Point Offset (Z) - Tool Properties window, 344
 - for mill tools, 157
 - for profile-type cutters, 160
 - for turn tools, 157
 - Mastercam converter, 250
 - Motion window, 315
- Control Report - Info menu function, 294
- Control Report file, 481
 - creating, 50
 - description of, 481
- Control settings, 365
- Control settings - see function name, 365
 - Circles, 365
 - Cutter Compensation, 375
 - Cycles, 366
 - General, 362
 - Motion, 363
 - Offsets, 376
 - OptiPath, 378
 - Rotary, 369
 - Subroutines, 377
 - Tooling, 368
 - User Defined, 380
 - Wire EDM, 374
- Control Type - General control settings window, 362
- Controls, 481
 - about building, 197
 - Control file, 481
 - Control Machine Simulation Speed - Motion window, 315
 - Control Report file, 481
 - Control Type - General control setting, 362
 - functions available for calling, 463
 - Image Playback features, 266
 - macros available for calling, 437
 - OptiPath menu function, 399
 - see also "Control file", 481
 - using MDI to test, 361
- Conventions - in CGTech documentation, 16
- Convert, 104
 - binary CL data, 111

Help on VERICUT

- G-codes to ASCII APT, 154
- IGES geometry, 104
- Mastercam NCI data, 112
- surface normals to be unidirectional, 109
- surfaces to solid models, 110
- VDA geometry, 107
- Converters, 19
 - Binary CL, 249
 - command line options for, 474
 - IGES, 240
 - Mastercam, 250
 - PolyFix, 243
 - Surface to Solid, 245
 - VDA, 247
- Coolant - Status window, 289
- COOLNT record, 508
- Coord. System - Setup menu, 336
- Coordinate System - View Axes window, 284
- Coordinate System Name - Coordinate System window, 336
- Coordinate systems and angles, 16
 - defining, 130
 - displaying axes of, 284
 - for view rotations, 57
 - general use in VERICUT, 16
- Copy, 176
 - component, 88
 - Copy all files - File Summary window, 292
 - copy files in use to a directory, 176
 - model, 97
 - optimization record in OptiPath Library, 402
 - tool in Tool Library, 172
- Copyright information, 1
- Corner Radius - Tool Manager mill tool, 338
- Create - see also "Convert", 110
- Create APT Output File - G-Code Output Files window, 324
- Cross Slide - Tool Manager turn tool profile, 338
- Cross Slide Center - Tool Manager turn tool profile, 338
- Cross-section - see "Section", 281
- cskdia - cycle def. mnemonic, 138
- CSS - see "constant surface speed", 318
- CSS Maximum RPM - G-Code Settings window
 - Turning tab, 318
- Curve Type - AUTO-DIFF window
 - Boundary tab, 394
- Cut - a component, 88
- Cut - a model, 97
- Cut Color list - Color window
 - Assign tab, 257, 508
- Cut Color Method - Motion window, 315
- Cut colors, 257
 - adding, 79
 - changing, 80
 - Cut Color Method - Motion window, 315
 - deleting, 79
 - list of, 257
 - PPRINT control of, 508
 - repainting, 80
- Cut Distance - OptiPath Manager window
 - Entry/Exit tab, 402
- Cut Mode, 261
 - choosing, 132
 - FastMill, 261
 - Properties window
 - General tab feature, 261
 - Standard, 261
 - Toolbar group, 285
- Cut Options - AUTO-DIFF window, 394
- Cut Seq. / Cut Color list - Color window
 - Assign tab, 257, 508
- cut sequence, 257
- Cuts - Section window, 281
- Cutter Change, 270
 - AutoSave window, 270
 - CUTTER record, 508
 - Image Playback - stop on, 266
 - Image Record window, 265
 - LOADTL record, 508
 - PPRINT control of, 508
 - see also "Tool Change", 311
- Cutter Compensation, 375
 - configuring control settings for, 211
 - debug messages for, 327
 - how VERICUT simulates, 420
 - Setup > Control > Settings function, 375
 - table, 420
- Cutter Height - Mastercam converter, 250
- CUTTER record, 160, 508
- Cutting Condition - OptiPath Manager
 - Feed/Speed tab, 402
- Cutting Tolerance - Properties window
 - Tolerance tab, 261
- cutting tools - see "Tools", 157
- CW Motion Cancels Cycles - Motion control settings window, 363
- cycle_op - cycle def. modal, 138
- Cycles, 366
 - About configuring for simulation, 138
 - APT-CLS record formats, 508
 - cancelling with word/addresses, 363, 368
 - configuring control settings for, 212
 - controlling display of, 136
 - Cycle Causes Rapid Motion, 366
 - Cycle Clearance Distance, 366
 - Cycle Definitions - APT Settings window, 331
 - Cycle Depth Value, 366
 - Cycle Execute, 366
 - Cycle Incremental Depth Value, 366
 - Cycle Incremental Rapid Value, 366
 - Cycle Rapid method, 366

Help on VERICUT

- Cycle Rapid Value, 366
- CYCLE record, 508
- Cycle Return Level, 366
- Cycle Set - APT Settings window, 331
- debug messages for, 327
- error messages and conditions, 674, 681
- Setup > APT Settings window tab, 331
- Setup > Control > Settings function, 366
- Cylinder, 298
 - Cylinder 1-2 - X-Caliper window
 - Scallop tab, 391
 - model type, 298
 - Tool Manager turn tool holder, 338
- CYLNRD record, 508
- D, 508
 - CUTTER record parameter, 508
 - Tool Manager mill tool parameter, 338
 - Tool Manager turn tool parameter, 338
- DataLink, 416
 - command line option, 467
 - OptiPath menu function, 416
- Debug, 327
 - configuring message output, 151
 - Debug Action Queue, 327
 - Debug Cutter Compensation, 327
 - Debug Cycle Values, 327
 - Debug Delta Information, 327
 - Debug Intersect Points, 327
 - Debug Loop Logic, 327
 - Debug Offsets, 327
 - Debug Rate Calculations, 327
 - Debug Substitution, 327
 - Debug Turning, 327
 - Debug Variables, 327
 - Debug Wire EDM, 327
 - Setup menu > G-Code > Process Options function, 327
- Deceleration - Modeling window
 - Component Attributes tab, 304
- deep_retract - cycle def. modal, 138
- Default, 363
 - Adv. OP Settings window
 - Spindle tab feature, 402
 - Default - Air Cut Feed Rate, 402
 - Default - Clean-up Feedrate, 402
 - Default - Maximum Cut Feedrate, 402
 - Default - Maximum RPM, 402
 - Default - Minimum Cut Feedrate, 402
 - Default - Minimum Feedrate Change, 402
 - Default - Minimum RPM, 402
 - Default - Minimum RPM Change, 402
 - Default CLEARP - APT Settings > Motion window, 331
 - Default Command Mode, 363
 - Default Control Units, 363
 - Default Feed Mode, 363
 - Default Feedrate, 363
 - Default Machine Mode, 362
 - Default Motion Type, 363
 - Default Near Miss - collision detection, 387
 - Default Normal - AUTO-DIFF, 394
 - Default Plane Selection, 363
 - Default Toolpath Units, 331
 - Default Units - Properties window
 - General tab, 261
 - Default Word - General control settings window, 362
 - Motion control settings, 214
 - OptiPath Manager window feature, 402
- Define - Color window tab, 257
- Delay - Section window, 281
- Delete, 67
 - collision detection case, 387
 - component, 88
 - coordinate system, 336
 - cut color, 79
 - Delete - OptiPath Manager window, 402
 - Delete- Coordinate System window, 336
 - Delete Material - Model menu, 310
 - display color, 77
 - event-driven macro or variable, 204
 - material, 101
 - model, 97
 - stored view, 75
 - subroutine file name, 149
 - substitution - from control config., 206
 - table or table row, 147
 - tool change event from tool list, 128
 - tool from Tool Library, 172
 - tool path files to simulate, 311
 - variable, 145
 - viewport, 67
 - word or special character, 200
- Word/Address class, 201
- Word/Address group, 202
- Depth Table - OptiPath Manager window tab, 402
- Depth/Feedrate table - OptiPath Manager window
 - Depth Table tab, 402
- Description - Status window, 289
- Description - Tool Add/Modify/Copy windows, 338
- Design - component type, 307
- Design Display - AUTO-DIFF window
 - Settings tab, 394
- Design Point - component type, 307
- Design Points file - description of, 481
- Detect 4-Axis Rotary Motion - APT Settings window
 - Rotary tab, 331

Help on VERICUT

Detect Chordal GOTOs - APT Settings window
Motion tab, 331
Dev Kit CME - Adv. Control Options window
tab, 381
Dev Kit CME File - Adv. Control Options
window
Dev Kit CME tab, 381
Developer's Kit - see "Machine Developer's Kit",
19
Diameter - Tool Manager mill tool, 338
Diameter - Tool Manager turn tool, 338
Diamond - Tool Manager turn tool, 338
Direction - APT Settings window
Rotary tab, 331
Direction - Tool Manager mill tool profile, 338
Disable Auto Direction - Modeling-Component
Attributes window, 304
Disabled - Image Record window, 265
Display - Toolbar group, 285
Display Resolution - Properties window
General tab, 261
Display Toolbar - View Toolbar window, 285
Distance, 391
Distance % - Status window, 289
Distance/Angle - X-Caliper window tab, 391
measuring between features, 180
Section window, 281
Status window, 289
DO/WHILE loops, 144
Documentation - CGTech products, 16
Done - Image Playback window, 266
Draw Mode, 279
changing, 70
view attribute, 279
Drill - Tool Manager mill tool, 338
Drill - Tool Manager turn tool, 338
Drill Cycle - Motion window, 315, 508
Drill Point Angle - Tool Manager mill tool, 338
Drill Point Angle - Tool Manager turn tool, 338
driven point - see "control point", 157
Driven Point Offset - Tool Manager-View
Attributes window, 338
Dynamic View - Toolbar group, 285
Dynamic View buttons - View Orient window,
277

E, 508
CUTTER record parameter, 508
E
E1
E2 - Tool Manager mill tool parameters,
338
Edit, 52
Edit menu - Component Tree window, 307
text file, 52
tool path file, 51

Embedded verification - for Pro/Engineer, 28
Encapsulated PostScript file - create from
VERICUT, 187
End - Section window, 281
Entry Feedrate - OptiPath Manager window
Entry/Exit tab, 402
Entry/Exit - OptiPath Manager window tab, 402
Environment variables, 479
adding to shortcuts, 479
formats for UNIX & Windows, 479
list of available, 479
EPSF file - create from VERICUT, 187
Error, 194, 674
collision with machine component, 193
color, 257
conditions - descriptions of, 681
Image Record window feature, 265
list of messages, 674
overtravel, 194
Status window field, 289
travel limit, 194
troubleshooting, 674
Event - Add/Modify Event windows, 381
Event - Adv. Control Options window
Events tab, 381
Event list (tool change) - Tool Change List
window, 311
Events - Adv. Control Options window tab, 381
Excess - AUTO-DIFF window
Tolerance tab, 394
Excess Check Distance - AUTO-DIFF window
Tolerance tab, 394
Excess Tolerance - AUTO-DIFF window
Tolerance tab, 394
Exit - the VERICUT program, 274
Exit Feedrate - OptiPath Manager window
Entry/Exit tab, 402
export - command line option, 467
Export Model, 252
command line option for, 467
exporting a model, 114
File menu function, 252
Extend to Rectangle - Modeling window
Import tab, 306
External File - Copy from Tool window, 338

F, 508
CUTTER record parameter, 508
F
F1
F2 - Tool Manager mill tool parameters,
338
Fadal control, 528
Fanuc control, 528
configuring cycles for, 366
imaginary tool nose programming, 375

Help on VERICUT

- library control files, 528
- setting the Control Type for, 362
- type C compensation, 420
- Fast Feed, 315, 508
 - color used to identify, 257
 - Fast Feed error - description of, 311, 674
 - Motion window feature, 315, 508
 - set feed rate value for, 135
- FastMill - cut mode, 261
 - PPRINT control of display, 508
- FastMill cut model - export from VERICUT, 114
- Feature options - X-Caliper window
 - Distance/Angle tab, 391
- Feature/History - X-Caliper window, 391
- Features - identified by X-Caliper, 177
- fedlist - cycle def. mnemonic, 138
- FEDRAT record, 508
- fedto - cycle def. mnemonic, 138
- fedupm - cycle def. mnemonic, 138
- Feed per Minute - OptiPath Manager window
 - Feed/Speed tab, 402
- Feed per Tooth - OptiPath Manager window
 - Feed/Speed tab, 402
- Feed/Speed - OptiPath Manager window tab, 402
- Feedrate - Status window, 289
- Feedrate Control Options - OptiPath Manager window
 - Depth Table tab, 402
- Feedrate Wear Reduction - OptiPath Manager window
 - Settings tab, 402
- Feedrate/Depth table - OptiPath Manager window
 - Depth Table tab, 402
- Fidia control, 528
- File - Export Model window, 252
- File - Toolbar group, 285
- File 1 SubSystem ID - G-Code Settings window
 - Turning tab, 318
- File 2 SubSystem ID - G-Code Settings window
 - Turning tab, 318
- File End - AutoSave window, 270
- file find logic, 42
- File Format - Export Model window, 252
- File menu, 229
 - Component Tree window, 307
 - Tool Manager window, 338
 - VERICUT main window, 229
- File Names list - Subroutines window feature, 208
- File Summary, 292
 - creating, 176
 - Info menu function, 292
- File Type - AUTO-DIFF window
 - Boundary tab, 394
- File Type - Export Model window, 252
- File types and descriptions, 481
- Filename - Image Playback window, 266
- Filename - Image Record window, 265
- Files - types used in VERICUT, 481
- Fill - OptiPath Manager window
 - Feed/Speed tab, 402
- Find - groups in a control config., 207
- Find (Word/Address) - Word/Address window, 357
- First - Find Word/Address window, 357
- Fit, 277
- Fit All, 277
 - all views - see "Fit All", 277
 - command line option, 467
 - on View Orient window, 277
 - via PPRINT/VERICUT-COMMAND record, 508
 - view, 64
- Fixed tool axis cycles - see "Cycles", 366
- Fixing STL or VERICUT model surface
 - normals, 109
- Fixture, 307
 - component type, 307
 - PPRINT control of display, 508
- Flat Bottom End Mill - Tool Manager mill tool, 338
- Flat Colors - Color window, 257
- floating zero location, 425
- Floor, 279
 - background attribute, 279
 - color assignment, 257
 - displaying, 72
 - setting "Machine Up" direction for, 387
- Font size - env. variable control of, 479
- Foreground - color definition, 257
- Forward - Tool Manager window, 338
- Forward & Backward - Image Record window, 265
- Forward Only - Image Record window, 265
- From, 299
 - Coordinate System window, 336
 - FROM record, 508
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position window, 347
 - X-Caliper window
 - Distance/Angle tab, 391
- Function Name - Add/Modify Word windows, 352
- Functions - listed by menu, 229
- Functions - map of 4.4 functions in 5.0, 222
- Functions (control) - listed alphabetically, 463
- G&L - see "Giddings & Lewis", 528
- G13 SubSystem ID - G-Code Settings window

Help on VERICUT

- Turning tab, 318
- G14 SubSystem ID - G-Code Settings window
 - Turning tab, 318
- Gage Length, 318
 - Gage Length (Z) - Tool Properties window, 344
 - gage length reference point, 157
 - programming method, 318
 - see also "Gage Offset", 422
- Gage Offset - table, 422
- Gage Offsets (X Y Z) - Tool Properties window, 344
- Gage Point Offset - Tool Manager-View
 - Attributes window, 338
- G-Code Log file, 481
- G-Code Log File - G-Code Output Files window, 324
 - creating, 156
 - description of, 481
 - viewing, 156
- G-Code Report - Info menu function, 293
- G-Code Report file, 481
 - creating, 50
 - description of, 481
- G-code tool path file - description of, 481
- GE control - see "General Electric", 528
- Gemini control, 528
 - configuring type of sub names for, 218
- General, 362
- General Electric control, 528
 - configuring General control settings, 213
 - Properties window tab, 261
 - Setup > Control > Settings function, 362
- Generic controls, 528
- geometric features, 177
- Geometry - Status window, 289
- Giddings & Lewis control, 528
- Go - see "Play to End", 40
- GODLTA record, 508
- GOHOME Motion - APT Settings window
 - Motion tab, 331, 508
- GOHOME R Dist - APT Settings window
 - Motion tab, 331
- GOHOME record, 508
- GOTO Format - APT Settings window
 - Turning tab, 331
- GOTO jumps, 144
- GOTO record, 508
- Gouge - AUTO-DIFF window
 - Tolerance tab, 394
- Gouge Check Distance - AUTO-DIFF window
 - Tolerance tab, 394
- Gouge Tolerance - AUTO-DIFF window
 - Tolerance tab, 394
- Graphics area, 35
- Groove - Tool Manager turn tool, 338
- Groups, 357
 - adding, 202
 - arranging into a class, 201
 - deleting, 202
 - finding, 207
 - modifying, 202
- Guide - component type, 307
- H, 508
- HEAD record, 508
- Heidenhain control, 528
 - setting a Default Word for, 362
 - setting the Control Type for, 362
- Height - Tool Manager mill tool, 338
- Height - Tool Manager turn tool, 338
- Height (Z) - Modeling window
 - Model tab, 298
- helical motion, 365
 - configuring control settings for, 210
- H-ISO - View Orient window, 277
 - CUTTER record parameter, 508
 - Tool Manager mill tool parameter, 338
 - Tool Manager turn tool parameter, 338
- History - of machine cuts, 177
- Holder Clearance - Motion window, 315
- Holders - using on tools, 125
- holdia - cycle def. mnemonic, 138
- Host - OptiPath Data Link window, 416
- ID - Status window, 289
- ID - Tool Add/Modify/Copy windows, 338
- IF/THEN checks, 144
- IGES converter, 240
 - command line options, 474
 - converting IGES data, 104
 - data converted by, 105
 - description of, 240
 - IGES version support, 105
 - window features, 240
- IGES data - supported by the IGES Converter, 105
- IGES model file, 481
 - converting, 104
 - description of, 481
 - exporting, 114
- IGES File - Modeling window
 - Import tab, 306
 - importing, 99
- IGES version- supported by the IGES Converter, 105
- Ignore - Mach. Settings window
 - Travel Limits tab, 387
- Ignore Collision between Cutter and Stock - Mach. Settings window
 - Collision Detect tab, 387
- Ignore Cycles during Rotary Motion - Cycles

Help on VERICUT

- control settings window, 366
- Ignore Data after Column - G-Code Settings window
 - Settings tab, 318
- ignoring columns of G-code data, 318
- IJK data fields - X-Caliper window, 391
- Image file, 481
 - description of, 481
 - playing back, 189
 - recording, 189
 - VCR controls, 266
- images - capturing in VERICUT, 185
- imaginary tool nose programming, 375
- Import, 306
 - IGES model, 99
 - importing and exporting data, 103
 - Modeling window tab, 306
- IMRCAPDT=on - command line option, 467
- In Process - AutoSave window tab, 270
- In Process File, 42, 481
 - autosaving, 270
 - command line option for, 467
 - description of, 481
 - opening, 47
 - saving, 47
- Inch Format - Add/Modify Word windows, 352
- Inch Method - Add/Modify Word windows, 352
- Inch/Millimeter option - OptiPath Manager window
 - Settings tab, 402
- Increment, 299
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position" window, 347
 - View Orient window, 277
- Incremental Circle Center Method - Circles control settings window, 365
- Index - Tables window feature, 191
- Info - Toolbar group, 285
- Info menu - VERICUT main window, 229
- Information - about the VERICUT session, 288
 - Control report, 294
 - files in use by VERICUT, 292
 - G-Code report, 293
 - Log file, 295
 - Results file, 296
 - Status window, 289
 - Toolpath file, 288
- init.usr/initm.usr - VERICUT initialization files, 232
- Initial Graphics Exchange Specification, 240
- Initial Machine Location table, 423
- Initial Tool - setup Toolpath window, 311
- Initial Tool Component ID - Tooling control settings window, 368
- Initial Voltage - Wire EDM control settings window, 374
- Initial Work Index - Offsets control settings window, 376
- Initial Work Offset - Offsets control settings window, 376
- Input File, 240
- Input File Type, 243
 - Binary CL converter, 249
 - IGES converter, 240
 - Mastercam converter, 250
 - PolyFix converter, 243
 - Surface to Solid converter, 245
 - VDA converter, 247
- Input Machine Zero table, 424
- Input Program Zero table, 425
- Input Text - Adv. Control Options window
 - Substitute tab, 381
- Input/Output Text list - Adv. Control Options window
 - Substitute tab, 381
- Inscribed Circle - Tool Manager turn tool, 338
- Insert - Component Tree-Component menu, 307
- Insert Angle - Tool Manager turn tool, 338
- Insert Delay - Image Record window, 265
- Insert Stop - Image Record window, 265
- Interactive/On/Off options - OptiPath Control window, 399
- Interpolated - Mach. Settings window
 - Axis Priority tab, 387
- Interpolation Tolerance - Properties window
 - Tolerance tab, 261, 365
- Introduction, 14
 - to VERICUT, 14
 - to VERICUT files, 42
- IP file - see "In Process file", 481
- ip= - command line option, 467
- Job file - description of, 481
- Job Tables - G-Code Settings window tab, 318
- JPEG file - create from VERICUT, 187
- Jump logic - see "Looping-branching logic", 327
- K&T - see "Kearney & Trecker", 528
- Kearney & Trecker control, 528
- Keep - material, 310
- KNOT record, 508
- Layout - views, 275
- Lead Angle - Tool Manager turn tool, 338
- Leblond Makino control, 528
- Left - ShadeCopy window
 - Postscript Settings tab, 268
- Length - Tool Manager turn tool, 338
- Length (X) - Modeling window
 - Model tab, 298

Help on VERICUT

- LGFILNAM=<logfilename> - command line option, 467
- Library files, 481
- Licenses, 18
 - VERICUT by CGTech, 19
 - VERICUT for Pro/Engineer, 28
- Light, 279
 - brightness, 83
 - direction - in Tool Manager, 338
 - direction - in VERICUT, 71
 - view attribute, 279
- Linear Motion Cancels Cycles - Motion control settings window, 363
- Linear-sweep file - description of, 481
- List Tool Change By - Tool Change List window, 311
- Load, 233
- LOAD/TOOL record, 508
 - Load - Image Playback window, 266
 - loading VERICUT files - see "Open", 233
- LOADTL record, 508
- Location - Coordinate System window, 336
- Log File, 42, 481
 - append to via command line option, 467
 - description of, 481
 - Info menu function, 295
 - interpreting errors in, 674
 - Log File - APT Output window, 324
 - naming, 295
 - naming - via command line option, 467
 - resetting - interactively, 55
 - resetting - via command line option, 467
 - viewing, 55
- Logical Sub-type - Add/Modify Word window, 352
- Look & Feel - View menu function, 287
- look ahead - CDC compensation, 420
- Loop - Image Playback window, 266
- looping-branching logic, 327
 - configuring scan pass for, 144
 - debug messages for, 327
- Machine Developer's Kit, 19, 328
 - configuring to use CME file from, 381
 - control settings created from, 380
 - description of, 19
 - G-code process options created from, 328
- Machine file, 481
 - about building, 190
 - description of, 481
 - library Machine files, 532
 - opening, 48
 - saving, 48
 - testing via MDI, 196
- Machine list - OptiPath Manager window, 402
- Machine Location - Modeling window
 - Position tab, 299
- Machine Origin - View Axes window, 284
- Machine Reference Location table, 427
- Machine Simulation, 19
 - CGTech license, 19
 - Machine Simulation On - Machine Settings window, 387
 - Pro/Engineer license, 28
- Machine Speed Control - Motion window, 315
- Machine Up - Machine Settings window, 387
- Machines, 190
 - about building, 190
 - detecting collisions, 193
 - initial machine location, 423
 - Machine - Modeling window
 - Component Attributes tab, 304
 - Machine - OptiPath Control window, 399
 - Machine Axes - Status window, 289
 - reference location, 427
 - using MDI to move, 361
- Machining Type - Motion window, 315
- Macro list - Add/Modify Event windows, 381
- Macro list - Add/Modify Word/Address windows, 357
- Macro/Variable list - Adv. Control Options window
 - Events tab, 381
- Macroname - Find Word/Address window, 357
- Macroname/Variable option, 357
 - Add/Modify Event window, 381
 - Add/Modify Word/Address window, 357
- Macros, 437
 - calling via event, 381
 - calling via word/address, 357
 - list of - in Word/Address window, 357
 - listed alphabetically, 437
- Maho control - see "Phillips", 528
- Makino control, 528
- Manager - OptiPath menu, 402
- Manual Data Input - see "MDI", 361
- manual tool change, 122
- Map - of 4.4 functions in 5.0, 222
- Margins - ShadeCopy window
 - Postscript Settings tab, 268
- Mastercam converter, 250
 - command line options, 474
 - converting NCI data, 112
 - description of, 250
 - Mastercam version support, 250
 - window features, 250
- Mastercam NCI data - convert to ASCII APT tool path, 112
- Material - OptiPath Control window, 399
- Material list - OptiPath Manager window, 402
- Math functions - see "Functions", 463
- Math Sub-type - Add/Modify Word windows,

Help on VERICUT

- 352
- Matrix - tool path, 331
 - PPRINT control of, 508
- Max Feed Velocity - Modeling window
 - Component Attributes tab, 304
- Max Tool Axis Angle - APT Settings window
 - Motion tab, 331
- Max Velocity for Direction Change - Modeling window
 - Component Attributes tab, 304
- Maximum - Mach. Settings window
 - Travel Limits tab, 387
- Maximum Cut Feedrate - OptiPath Manager window
 - Settings tab, 402
- Maximum Inverse Time Output Feedrate - Motion control settings window, 363
- Maximum RPM - OptiPath Manager window
 - Spindle tab, 402
- Maximum Wire Angle - Wire EDM control settings window, 374
- Maximum XYZ - AUTO-DIFF window
 - Tolerance tab, 394
- Mazak control, 528
- MDI - Setup menu > Control function, 361
- Measure, 177
 - air distance, 184
 - distance or angle between 2 features, 180
 - from a point to closest point on the model, 182
 - geometric features, 177
 - history info, 177
 - scallop height, 181
 - thickness of solid material, 183
- Menu area, 36
- merged turning, 318
- merging a User file, 46
- Message area, 38
- Messages - errors and warnings, 674
- Messages - Setup menu > G-Code > Process
 - Options function, 325
- Metal - Look & Feel option, 287
- Method - ShadeCopy window
 - Attributes tab, 268
- Metric Format - Add/Modify Word windows, 352
- Metric Method - Add/Modify Word windows, 352
- Mill tools, 157
 - about describing, 157
 - control point, 157
 - describing in Tool Manager, 338
 - mill tool components, 157
- Millimeter/Inch option - OptiPath Manager window
 - Settings tab, 402
- Milling, 315
 - machining type, 315
 - setting Default Machine Mode for, 362
- Min. Cutter Height - Motion window, 315
- Minimum Area - AUTO-DIFF window
 - Boundary tab, 394
- Minimum - Mach. Settings window
 - Travel Limits tab, 387
- Minimum Cut Feedrate - OptiPath Manager window
 - Settings tab, 402
- Minimum Cutter Height - Binary CL converter, 249
- Minimum Edge - Export Model window, 252
- Minimum Error Volume - Properties window
 - Tolerance tab, 261
- Minimum Feedrate Change - OptiPath Manager window
 - Settings tab, 402
- Minimum Gap - AUTO-DIFF window
 - Boundary tab, 394
- Minimum RPM - OptiPath Manager window
 - Spindle tab, 402
- Minimum RPM Change - OptiPath Manager window
 - Spindle tab, 402
- Minimum XYZ - AUTO-DIFF window
 - Tolerance tab, 394
- Mixed Mode - Modeling window
 - Component Attributes tab, 304
- Modals - see "Control settings", 365
- MODE record, 508
- Models, 297
 - about modeling in VERICUT, 85
 - assembling, 95
 - checking consistency of stock, 102
 - defining, 297
 - displaying axes of, 284
 - fitting in view, 64
 - importing, 99
 - Model - Modeling window tab, 298
 - Model - View Axes window, 284
 - Model Coordinates - AUTO-DIFF window
 - Boundary tab, 394
 - Model Coordinates - Export Model window, 252
 - Model Definition - Model menu, 297
 - Model Export - license, 19
 - Model File - model type, 298
 - Model menu - VERICUT main window, 229
 - Model Tolerance - Properties window
 - Tolerance tab, 261
 - moving, 95
 - paning view of, 63
 - positioning, 95
 - PPRINT control of display, 508
 - reversing view of, 61

Help on VERICUT

- rotating, 95
- rotating view of, 59
- sample session assembling, 554
- sample session defining, 544
- tolerance for revolved shapes, 261
- zooming in/out on, 62
- Modify, 74
 - collision detection case, 387
 - component, 88
 - event-driven macro or variable, 204
 - model, 97
 - Modify Existing Cuts Only - OptiPath Manager window
 - Settings tab, 402
 - stored view, 74
 - substitution, 206
 - table or table row, 147
 - tool change event in a tool list, 128
 - tool in Tool Library, 172
 - travel limits, 194
 - variable, 145
 - view name, 74
 - word or special character, 200
 - Word/Address class, 201
 - Word/Address group, 202
- Monitoring the simulation, 173
- Motif - Look & Feel option, 287
- Motion, 315
 - APT Settings window tab, 331
 - configuring control settings for, 214
 - features on Motion window, 315
 - Motion Axis - Modeling window
 - Component Attributes tab, 304
 - PPRINT control of, 508
 - Setup > Control > Settings function, 363
 - Setup menu function, 315
- mouse pick mode - for sectioning, 281
- MOVARC record, 508
- Move, 299
- Move/Rotate - Tool Manager "Position" window, 347
 - component or model, 95
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - tool holder, 347
 - Tool Manager "Position" window, 347
- MSYS record, 508
- MULTAX - APT Settings window
 - Motion tab, 331
- MULTAX record, 508
- Multi-Axis - license, 19
- Multiplier - Add/Modify Word windows, 352
- Multiply - Add/Modify Word windows, 352
- N Block SubSystem ID - G-Code Settings
 - window
 - Turning tab, 318
- N Records - Image Record window, 265
- Name, 352
 - component, 93
 - G-code word, 352
 - see also "Rename", 93
 - variable, 318
 - Word/Address class, 201
- NC macros - see "Subroutines", 318
- NC Optimization - license, 28
- NC Verification - license, 28
- NCI data - convert to ASCII APT tool path, 112
- Neg/Pos - Mach. Settings window
 - Axis Priority tab, 387
- Negative/Positive options - Mach. Settings window
 - Axis Priority tab, 387
- New, 232
 - File menu function, 232
 - OptiPath Library, 402
 - Tool Library, 168
 - VERICUT session, 44
- Next - Find Word/Address window, 357
- None - Tool Manager mill tool, 338
- Normal Direction, 240
 - AUTO-DIFF window
 - Tolerance tab, 394
 - IGES converter, 240
 - VDA converter, 247
- Normal I-Normal J-Normal K - Section window, 281
- Normals - Modeling window
 - Model tab, 298
- Normals - Tool Manager turn tool, 338
- Nose Diameter - Tool Manager turn tool, 338
- Nose Radius - Tool Manager turn tool, 338
- NUM control, 528
 - library control files, 528
 - setting the Control Type for, 362
- Number - Status window, 289
- NumeriPath control, 528
- NURBS, 508
 - NURBS Feedrate - OptiPath Manager window
 - Settings tab, 402
 - NURBS Only - Export Model window, 252
 - NURBS record, 508
- O Block SubSystem ID - G-Code Settings window
 - Turning tab, 318
- Off/On/Interactive options - OptiPath Control window, 399
- Offset, 245
 - Auto Block on Modeling window
 - Model tab, 298

Help on VERICUT

- component or model constraint, 301
- Offset Amount - Modeling window
 - Import tab, 298
- Offset Surface - Modeling window
 - Import tab, 306
- Surface to Solid converter, 245
- tool holder constraint, 347
- Offsets, 376
 - configuring control settings for, 215
 - Setup > Control > Settings function, 376
- Okuma control, 528
- Olivetti control, 528
- On - AutoSave window
 - In Process tab, 270
 - ShadeCopy tab, 270
- On/Off/Interactive options - OptiPath Control window, 399
- OP Feedrate - Status window, 289
- OP Spindle - Status window, 289
- OP Time - Status window, 289
- Open, 46, 47
 - Control file, 49, 349
 - Image file, 189, 266
 - In Process file, 47, 236
 - Machine file, 48, 307, 384
 - OptiPath Library file, 402
 - Tool Library file, 169
 - Toolbar, 37
 - User file, 46, 233
- OpenGL - command line option, 467
- Optimization Method - OptiPath Manager window
 - Feed/Speed tab, 402
- optimize - command line option, 467
- Optimize by Tables - optimization method, 402
- Optimized File, 481
 - description of, 481
 - OptiPath Control window, 399
- Optimizing toolpaths, 19
 - about, 221
 - controlling block sequencing and spacing, 216
 - via interactive prompting, 311
- Options - Toolbar group, 285
- OptiPath, 19
 - command line option, 467
 - configuring control settings for, 216
 - Control function, 399
 - DataLink function, 416
 - license description, 19
 - light - on VERICUT main window, 41
 - Manager function, 402
 - optimizing multi-axis motions, 369
 - OptiPath Description - Tool Properties window, 344
 - OptiPath Library - OptiPath Control window, 399
 - OptiPath Library file - description of, 481
 - OptiPath menu - VERICUT main window, 229
 - OptiPath tool properties, 344
 - PPRINT control of, 508
 - Setup > Control > Settings function, 378
- Order of Math Operations - General control settings window, 362
- Orient - Coordinate System window, 336
- Orient - Tool Manager > View menu, 338
- Orient - View menu, 277
- Orientation - ShadeCopy window
 - Attributes tab, 268
- Orientation list - setup Toolpath window, 311
- Other - component type, 307
- Output - AUTO-DIFF window
 - Boundary tab, 394
- Output - Export Model window, 252
- Output - ShadeCopy window, 268
- Output "Word is not defined" Errors - Messages window, 325
- Output Block Sequence Numbers - Messages window, 325
- Output Circles - Binary CL converter, 249
- Output File, 240
- Output File Format, 240
 - IGES converter, 240
 - PolyFix converter, 243
 - Surface to Solid converter, 245
- Output File Type, 240
 - APT Output File - G-Code Output Files window, 324
 - AUTO-DIFF window
 - Boundary tab, 394
 - Binary CL converter, 249
 - IGES converter, 240
 - Mastercam converter, 250
 - PolyFix converter, 243
 - ShadeCopy window
 - Attributes tab, 268
 - Surface to Solid converter, 245
 - VDA converter, 247
- Output Filenames Format - Messages window, 325
- Output Files - Setup menu > G-Code > Process Options function, 324
- Output Intermediate Points - Rotary modals window, 369
- Output Initial Spindle Location - Motion control settings window, 363
- Output Precision - Export Model window, 252
- Output Precision - Motion control settings window, 363
- Output Rapid Motions as - Motion control settings window, 363
- Output Resolution - Export Model window, 252

Help on VERICUT

- Output Text - Adv. Control Options window
 - Substitute tab, 381
- Override, 124
 - current tool, 124
 - Override All Feedrates - OptiPath Control window, 399
 - Override Text - Add/Modify Event windows, 381
 - Override Text - Add/Modify Word/Address windows, 357
 - Override Value - Add/Modify Event windows, 381
 - Override Value - Add/Modify Word/Address windows, 357
- Overtravel Color - Mach. Settings window
 - Travel Limits tab, 387
- Overtravel Detection On - Mach. Settings window
 - Travel Limits tab, 387
- Overtravel errors - how to detect, 194
- Overview, 31
 - of using VERICUT, 31
- Pan, 277
 - on View Orientation window, 63
 - view, 63
- Paper Size - ShadeCopy window
 - Postscript Settings tab, 268
- parsing - G-code word values, 352
- Part edge - programming method, 420
- Part Transformation table, 428
- Paste, 307
 - component, 88
 - model, 97
- pause - PPRINT control of, 508
- pbappend - command line option, 467
- Phillips control, 528
 - configuring base work offset for, 419
 - configuring CDC for, 375
 - configuring cycles for, 366
 - library control files, 528
- Philosophy - APT Settings window
 - Rotary tab, 331
- Pitch Specified with IJK - Circles control settings window, 365
- Plane 1-Plane 2-Plane 3 - Section window, 281
- Plane option list - Section window, 281
- Plane Type - Section window, 281
- Play to End - simulation control (VCR button), 40
- Playback - File menu > Images function, 266
- Playback control VCR icons - Image Playback window, 266
- Plunge - OptiPath Manager window tab, 402
- Plunge Feedrate - OptiPath Manager window
 - Plunge tab, 402
- Point - Tool Manager mill tool profile, 338
- Point - Tool Manager turn tool profile, 338
- Poly model files - see "VERICUT model files", 481
- PolyFix converter, 243
 - command line options, 474
 - description of, 243
 - fixing model normals, 109
 - window features, 243
- Port - OptiPath Data Link window, 416
- Pos/Neg - Mach. Settings window
 - Axis Priority tab, 387
- Position, 299
 - component or model, 95
 - coordinate system, 130
 - Coordinate System window, 336
 - Modeling window tab, 299
 - Tool Add/Modify/Copy windows, 347
 - tool holder, 347
 - Tool Manager "Position" window, 347
 - tool path, 130
- Positive/Negative options - Mach. Settings window
 - Axis Priority tab, 387
- post-processing, 481
- PostScript file - create from VERICUT, 187
- Postscript Settings - ShadeCopy window tab, 268, 481
- PPRINT, 508
 - G-code equivalent format, 508
 - PPRINT record (general purpose), 508
 - PPRINT TOOL AXIS record, 508
 - PPRINT/ATP CBOX record, 508
 - PPRINT/VERICUT record, 508
 - PPRINT/VERICUT-COMMAND record, 508
 - PPRINT/VERICUT-CUTCOLOR record, 508
 - PPRINT/VERICUT-MATRIX record, 508
 - PPRINT/VERICUT-MODAL record, 508
 - PPRINT/VERICUT-MOTION record, 508
 - PPRINT/VERICUT-OPTIPATH, 508
 - PPRINT/VERICUT-TC record, 160, 508
- Preferences file, 481
 - controlling name of, 479
 - description of, 481
 - resetting, 481
- Prefs file - see "Preferences file", 481
- Previous - Find Word/Address window, 357
- primary axes - 4-axis turning machines, 318
- Primary SubSystem ID - G-Code Settings window
 - Turning tab, 318
- Print, 264
 - File menu function, 264
 - Print Command - AutoSave window
 - ShadeCopy tab, 270
 - printing VERICUT images, 186

Help on VERICUT

- Priority - Mach. Settings window
 - Axis Priority tab, 387
- Problem solving, 674
- Process, 240
 - Binary CL converter, 249
 - Export Model window, 252
 - IGES converter, 240
 - Mastercam converter, 250
 - PolyFix converter, 243
 - Process After Motion - Add/Modify Word/Address windows, 357
 - Process Circles - APT Settings window
 - Motion tab, 331, 508
 - Process Cutter Comp. - G-Code Settings window
 - Settings tab, 318
 - Process during Scan - Add/Modify Word/Address windows, 357
 - Process Matrix - APT Settings > Motion window, 331, 508
 - Surface to Solid converter, 245
 - VDA converter, 247
- Processing Complete - Image Record window, 265
- Processing controls, 40
- Profile, 338
 - defining tool profiles, 160
 - examples of defining, 160
 - Tool Manager mill tool, 338
 - Tool Manager turn tool, 338
- Programming Method, 318
 - "part edge", 420
 - gage length, 318
 - G-Code Settings window
 - Settings tab feature, 318
 - setting, 144
 - tool length compensation, 318
 - tool tip, 318
- Progress bar, 39
- Project Rectangle - Surface to Solid window, 245
- Project to Plane - AUTO-DIFF window
 - Boundary tab, 394
- Prompt for OptiPath Settings - Tool Change List window, 311
- Properties, 261
 - AutoSave window
 - ShadeCopy tab feature, 270
 - File menu function, 261
 - for tools defined via Tool Manager, 344
 - Tool Add/Modify/Copy windows feature, 338
 - verification session, 261
- Protect Cuts - AUTO-DIFF window
 - Cut Options tab, 394
- prshade - command file, 270
- PS file - see "PostScript", 187
- pseudocolor - command line option, 467
- R, 508
 - CUTTER record parameter, 508
- R
 - R1
 - R2 - Tool Manager mill tool parameters, 338
 - Tool Manager turn tool parameter, 338
- Radial Width - OptiPath Manager window
 - Feed/Speed tab, 402
- Radius, 298
 - Modeling window
 - Model tab, 298
 - Tool Manager mill tool profile, 338
 - Tool Manager turn tool, 338
 - X-Caliper window
 - Scallop tab, 391
- Range - Add/Modify Word/Address windows, 357
- Range - Find Word/Address window, 357
- Range/Color list - AUTO-DIFF window
 - Tolerance tab, 394
- Rapid Motion, 387
 - configuring for APT-CLS tool path, 331
 - configuring for machine, 195
 - RAPID Motion - APT Settings window
 - Motion tab, 331
 - Rapid Motion Cancels Cycles - Motion control settings window, 363
 - Rapid Rate - of machine axis, 304
- Rapid prototyping, 19
- Rapid Rate - Modeling window
 - Component Attributes tab, 304
- RAPID record, 508
- rapto - cycle def. mnemonic, 138
- Recent Files - File menu function, 273
- Record, 265
 - AVI file, 56
 - button on Image Record window, 265
 - command line option, 467
 - File menu > Images function, 265
 - Image file, 56
 - VERICUT images (snapshots or animation), 56
- Recycle Cut Color - Color window
 - Assign tab, 257
- Red-Green-Blue slidebars - Color window
 - Define tab, 257
- Redraw - View Orientation window, 277
- Reference, 338
 - Reference location - table for defining, 427
 - Tool Manager mill tool, 338
 - Tool Manager turn tool, 338
- Refine Display - Toolbar option, 285
- Register - G-Code Settings window

Help on VERICUT

- Tool Tables tab, 318
- Register - Tables window feature, 191
- Relative Location - Modeling window
 - Position tab, 299
- Rename, 74
 - component, 93
 - stored view, 74
 - Word/Address class, 201
- Repaint - cut colors, 80, 257
 - PPRINT control of, 508
- Repairing STL or VERICUT model file normals, 109
- Replace, 318
 - see also "Substitute", 381
 - subroutine file name, 149
 - tool path file name, 311
- Report - AUTO-DIFF window, 394
- Report file, 481
 - Control - codes control can handle, 481
 - G-Code - codes in TP and how control will handle, 481
- Requirements - for a simulation, 31
- Reset, 40
- Reset Cut - color for resetting cut colors, 257
- Reset Cut Color, 257
 - command line option, 467
 - on Color window
 - Assign tab, 257
 - PPRINT control of, 508
 - resetting cut colors, 81
- Reset Model - simulation control, 40
- Reset Tool Change - APT Settings window
 - Motion tab, 331
 - cut colors, 81
 - Log file, 55, 467
 - model, 40
 - Modeling window
 - Assemble tab, 301
 - on Select/Store View window, 283
 - Preferences file settings, 481
 - simulation control (VCR button), 40
 - view orientation, 277
- reset_log - command line option, 467
- Resize - the VERICUT window, 34
- Resolution - OptiPath Manager window
 - Settings tab, 402
- Restore - AUTO-DIFF window, 394
- Restore - Section window, 281
- results - command line option, 467
- Results - Properties window tab, 261
- Results File, 42, 481
 - creating - interactively, 56
 - creating - via command line option, 467
 - description of, 481
 - Info menu function, 296
 - viewing, 56
- Retract Feedrate - OptiPath Manager window
 - Plunge tab, 402
- Retract for tool change, 431
- retract_mode - cycle def. modal, 138
- RETRCT record, 508
- Reverse, 61
- Reverse post-processing, 154, 324
 - command line option, 467
 - how to, 154
 - PPRINT control of, 508
 - Reverse Circles - APT setting, 331
 - Reverse Direction - component motion
 - direction, 304
 - view, 61
 - viewing APT output from, 155
- Rewind Toolpath - simulation control, 40
- Right - ShadeCopy window
 - Postscript Settings tab, 268
- ROTABL - APT Settings window
 - Rotary tab, 331, 508
- ROTABL record, 508
- Rotary, 331
 - APT Settings window tab, 331
 - APT-CLS record formats, 508
 - configuring control settings for, 217
 - Rotary Feed Motion Tol - G-Code Settings window
 - Settings tab, 318
 - Rotary Rapid Motion Tol - G-Code Settings window
 - Settings tab, 318
 - Rotary Tool Control Point - Rotary control settings window, 369
 - Setup > Control > Settings function, 369
- Rotate, 59
 - component or model, 95
 - coordinate system, 130
 - Coordinate System window, 336
 - dynamic rotation controls, 277
 - Modeling window
 - Position tab, 299
 - ROTATE - APT Settings window
 - Rotary tab, 331, 508
 - ROTATE record, 508
 - tool holder, 347
 - Tool Manager "Position" window, 347
 - tool path, 130
 - view, 59
- Rotation Angle - Tool Properties window, 344
- Rotation buttons - View Orient window, 277
- Rotation Center - APT Settings window
 - Rotary tab, 331
- ROTHED - APT Settings window
 - Rotary tab, 331, 508
- ROTHED record, 508
- RPM Changes with RAPID and CSS - G-Code

Help on VERICUT

- Settings window
 - Turning tab, 318
- RTCP, 369
 - Rotary Tool Control Point - Rotary control setting, 369
 - RTCP Pivot Offset table, 429
- rtcrto - cycle def. mnemonic, 138
- Sample Distance - OptiPath Data Link window, 416
- Sample files, 481
- Save, 46, 47
 - Control file, 49, 351
 - EPSF image file, 187
 - Image file, 189, 265
 - In Process file, 47, 238
 - JPEG image file, 187
 - Machine file, 48, 307, 386
 - OptiPath Library file, 402
 - PostScript image file, 187
 - ShadeCopy images, 187
 - TIF image file, 187
 - Tool Library file, 169
 - User file, 46, 235
- Scallop - X-Caliper window, 391
- scallop height - how to measure, 181
- scan pass, 318, 381
 - configuring for, 144
 - processing macros during, 357
- Scan Toolpath Files - G-Code Settings window
 - Settings tab, 318
- Search - Tool Manager window, 338
- secondary axes - 4-axis turning machines, 318
- Secondary SubSystem ID - G-Code Settings window
 - Turning tab, 318
- Section, 281
 - automatic, 281
 - how to, 65
 - restoring from, 65
 - Section on mouse pick - Section window, 281
 - via mouse pick, 281
 - View menu function, 281
- Select/Store - View menu function, 283
- Selected - Modeling window, 297
- Send Debug Messages - Messages window, 325
- Send Error Messages - Messages window, 325
- Send Info Messages - Messages window, 325
- Send Input Messages - Messages window, 325
- Send Modal Messages - Messages window, 325
- Send Output Messages - Messages window, 325
- Send Warning Messages - Messages window, 325
- Sequence - Status window, 289
- Sequence Increment Value - OptiPath control settings window, 378
- Session units, 34
- Set a working directory, 45
- Set All - Mach. Settings window
 - Collision Detect tab, 387
- Set All - View Axes window, 284
- Set All - View Toolbar window, 285
- SET/MODE record, 508
- Setting environment variables, 479
- Settings, 318
 - AUTO-DIFF window, 394
 - Control settings - see setting name, 365
 - G-code settings, 318
 - OptiPath Manager window tab, 402
 - Settings - G-Code Settings window tab, 318
 - Setup menu > G-Code function, 318
 - Setup menu > Machine function, 387
- Setup - Toolbar group, 285
- Setup menu - VERICUT main window, 229
- Shade Color - Color window
 - Define tab, 257
- Shade Type - Tool Manager-View Attributes window, 338
- Shade Type - View Attributes window, 279
- ShadeCopy, 268
 - autosaving, 270
 - configuring, 268
 - File menu > Images function, 268
 - outputting images from, 187
 - ShadeCopy - AutoSave window tab, 270
 - ShadeCopy File - AutoSave window
 - ShadeCopy tab, 270
- Shading By - ShadeCopy window
 - Attributes tab, 268
- Shortcut, 42
 - adding via env. var, 479
 - in file selection windows, 42
- Siemens control - see "Sinumerik", 528
- Simulation control VCR icons, 40
- Simulation speed - see "Animation Speed", 175
- Simultaneous Machine Simulation - Image
 - Playback window, 266
- Single - Mach. Settings window
 - Axis Priority tab, 387
- Single block - see "Single step", 40
- Single Step - simulation control (VCR button), 40
- Sinumerik control, 528
- Skip Cut - Motion window, 315
- Skip Duplicate Cutter Descriptions - Tool
 - Change List window, 311
- skipping blocks - how to configure, 152
- SLA file - see "STL file", 481
- slice - command line option, 467
- Snapshot, 265
 - playing back, 189
 - recording, 189

Help on VERICUT

- Snapshot - Image Record window, 265
- Snapshot Delay - Image Playback window, 266
- Snapshot Events - Image Record window, 265
- Solid models - create from surfaces, 110
- Solid-of-revolution file - description of, 481
- SOR file - description of file, 481
- special cutters, 160
- Special Sub-type - Add/Modify Word windows, 352
- Specified Output Precision - Motion control settings window, 363
- Specify Temporary Cutter Shape - OptiPath Manager window, 402
- Speed, 175
 - controlling animation speed, 175
 - of animation, 315
 - of Image playback, 266
- SPINDL record, 508
- Spindle, 307
 - component type, 307
 - OptiPath Manager tab, 402
 - Spindle Power - OptiPath Manager window Feed/Speed tab, 402
 - Spindle Speed - optimization method, 402
 - Status window field, 289
- Standard - cut mode, 261
- Standard - View menu > Layout, 275
- Standard Drawing View buttons - View Orient window, 277
- Start - OptiPath Data Link window, 416
- Start - Section window, 281
- Start - VERICUT processing, 40
- Start At Line - Motion window, 315
- Static View - Toolbar group, 285
- Static View buttons - View Orient window, 277
- Status, 289
 - Info menu function, 289
 - on simulation an machine conditions, 54
 - Status file - description of, 481
 - viewing, 54
- step - cycle def. mnemonic, 138
- steplist - cycle def. mnemonic, 138
- steprap - cycle def. mnemonic, 138
- Stereolithography - see "STL", 481
- STL file, 481
 - creating from IGES data, 104
 - creating from VDA data, 107
 - description of, 481
 - exporting, 114
 - fixing normals, 109
 - modeling with, 97
 - STL File - Tool Manager turning tool holder, 338
- Stock - component type, 307
- Stock Consistency Check - Properties window
 - General tab, 261
- Stock Display - AUTO-DIFF window
 - Settings tab, 394
- Stock Thickness - X-Caliper window, 391
- Stop, 40
 - APT-CLS record, 508
 - image playback, 266
 - Image Record window feature, 265
 - on errors or warnings, 174
 - on warnings, 174
 - OptiPath Data Link window feature, 416
 - Simulation control (VCR button), 40
 - Stop At - stop on events (e.g. tool chg text etc.), 174
 - Stop At Max Errors, 315
 - Stop At Max Warnings, 315
- Store - custom views, 73
- Subroutines, 318
 - adding sub file name, 149
 - Adv. Control Options window tab, 381
 - configuring access to, 208
 - configuring control settings for, 218
 - configuring scan pass for, 144
 - control subroutines, 208
 - deleting sub file name, 149
 - G-Code Settings window tab, 318
 - job subroutines, 208
 - levels of nesting, 208
 - renaming sub file name, 149
 - Setup > Control > Settings function, 377
 - types, 208
- Substitute - Adv. Control Options window tab, 381
- Substitution list - Adv. Control Options window Substitute tab, 381
- Substitutions, 381
 - debug messages for, 327
- Subsystems, 304
 - in a machine, 304
 - SubSystem - Modeling window
 - Component Attributes tab, 304
 - SubSystem - Status window feature, 289
 - SubSystem ID - Mach. Settings window
 - Tables tab, 387
 - SubSystem ID - Tables window feature, 191
- Summary - IGES converter, 240
- Summary of files in use, 176
- Sundstrand control, 528
- supergroups, 197
- Surface Input Mode - IGES converter, 240
- SURFACE record, 508
- Surface Speed - OptiPath Manager Feed/Speed tab, 402
- Surface to Solid converter, 245
 - command line options, 474

Help on VERICUT

- converting surfaces to solids, 110
 - description of, 245
 - window features, 245
 - Switch 1-9 - G-Code Settings window
 - Block Skip tab, 318
 - SWP file - description of, 481
 - synchronization codes - 4-axis turning machines, 318
 - Table Name - Tables window feature, 191
 - Tables, 417
 - about configuring tables (GUI features), 191
 - adding, 147
 - Base Work Offset, 419
 - Cutter Compensation, 420
 - deleting, 147
 - Gage Offset, 422
 - Initial Machine Location, 423
 - Input Machine Zero, 424
 - Input Program Zero, 425
 - listed alphabetically, 417
 - Machine Reference Location, 427
 - modifying, 147
 - Part Transformation, 428
 - RTCP Pivot Offset, 429
 - Tables - Advanced OptiPath Settings window feature, 402
 - Tables - G-Code Settings window tabs, 318
 - Tables - Machine Settings window tabs, 387
 - Tool Change Location, 430
 - Tool Change Retraction, 431
 - Tool Length Compensation, 432
 - Tool Nose Compensation, 433
 - Work Offset, 435
 - Tag Image Format file - create from VERICUT, 187
 - Text File, 256
 - editing, 52
 - File menu > Edit function, 256
 - thickness - measure solid material, 183
 - Thickness - Tool Manager turn tool, 338
 - thru_mode - cycle def. modal, 138
 - TIF file - create from VERICUT, 187
 - Tile Horizontally - view, 275
 - Tile Vertically - view, 275
 - Time - Status window, 289
 - Time % - Status window, 289
 - Time Interval - Status Configuration window, 289
 - Tip 1/2 Angle - Tool Manager turn tool, 338
 - tlangl - cycle def. mnemonic, 138
 - TLAXIS record, 508
 - TLDATA record, 508
 - To, 299
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position" window, 347
 - X-Caliper window
 - Distance/Angle tab, 391
 - tokens, 197
 - Tolerance, 240
 - AUTO-DIFF tolerance, 394
 - Circle Radial Tolerance, 261
 - collision detection, 387
 - Cutting Tolerance, 261
 - for calculating circle center to start/end points, 261
 - for cuts in the workpiece, 261
 - for importing IGES model files, 306
 - for interpolated motions, 261
 - for math equations in G-codes, 362
 - for modeling cylindrical and revolved solids, 261
 - for Tool Manager turn tool, 338
 - IGES converter, 240
 - Interpolation Tolerance, 261
 - Minimum Error Volume, 261
 - Model Tolerance, 261
 - Properties window tab, 261
 - Rotary Feed Motion Tol, 318
 - Rotary Rapid Motion Tol, 318
 - Surface to Solid converter, 245
 - VDA converter, 247
- Tool - component type, 307
- Tool - see "Tools", 157
- Tool axis cycles - see "Cycles", 366
- Tool Axis Reset Angle - APT Settings window
 - Motion tab, 331
- Tool Change, 311, 508
 - Event list (tool change) - Tool Change List window, 311
 - methods of changing, 118
 - see also "Change tools", 118
 - Tool Change By - on setup Toolpath window, 311, 508
 - Tool Change Cancels Cycles, 368
 - Tool Change Causes Rapid Motion, 368
 - Tool Change Location table, 430
 - Tool Change Retract Height - Mastercam converter, 250
 - Tool Change Retract Method, 368
 - Tool Change Retraction table, 431
 - tool path records, 508
 - using tool holders, 125
 - via PPRINT/VERICUT-TC record, 508
- Tool Component - Tool Add/Modify/Copy windows, 338
- Tool component shape icons - Tool Add/Modify/Copy windows, 338
- Tool Description list - OptiPath Manager window, 402

Help on VERICUT

- Tool Display - Motion window, 315
- Tool holders - controlling use of, 125
- Tool Index - Modeling window
 - Component Attributes tab, 304
- Tool Length Compensation, 318
 - programming method, 318
 - table, 432
- Tool Library, 481
 - adding a tool to, 170
 - creating a new library, 168
 - sample session creating, 594
 - Tool Library file - description of, 481
 - Tool Manager used to create, 338
- Tool list, 126
 - about using tool lists, 126
 - adding event to, 128
 - building, 127
 - deleting event from, 128
 - modifying event in, 128
 - sample session using, 608
- Tool Manager - Setup menu function, 338
- Tool Nose Compensation table, 433
- Tool Override - setup Toolpath window, 311
- Tool path - see also "Toolpath", 481
- Tool path records - APT-CLS, 508
- Tool properties, 344
 - mill tools, 344
 - turn tools, 344
- Tool Tables - G-Code Settings window tab, 318
- Tool Tip, 289
 - programming method, 318
 - Status window fields, 289
- Tool Type - Tool Add/Modify/Copy windows, 338
- Tool Use graph - Status window, 289
- Tool/ID copy list - Copy from Tool window, 338
- Tool/ID list - Tool Manager window, 338
- Toolbar, 37
 - closing, 37
 - customizing, 37
 - functions accessed by, 285
 - opening, 37
 - View menu function, 285
- Tooling, 368
 - configure control settings for, 219
 - Setup > Control > Settings function, 368
- Toolpath - Info menu function, 288
- Toolpath - Setup menu function, 311
- Toolpath File, 42, 481
 - default units, 331
 - description of, 481
 - editing, 51
 - File menu > Edit function, 255
 - locating via coord. systems, 130
 - sample session- multiple setups
 - same toolpath, 575
 - sample session- multiple toolpaths
 - different orientations, 583
 - same orientation, 579
 - specifying, 116
 - viewing, 53
- Toolpath Record - Status window, 289
- Toolpath Type - setup Toolpath window, 311
- Toolpath/Coord. System list - setup Toolpath window, 311
- Tools, 157
 - changing, 118
 - color of, 78
 - color of cuts by, 79
 - controlling display of, 315
 - descriptions via Tool Manager, 170
 - descriptions via tool path records, 508
 - overriding current tool, 124
 - sample session defining tool profiles, 597
 - setting an initial tool, 123
 - Tool Number Method - Tooling control settings window, 368
 - Tool Angle - Tool Manager turn tool, 338
 - Tool Assembly - Tool Add/Modify/Copy windows, 338
- Top - ShadeCopy window
 - Postscript Settings tab, 268
- Top Radius, 298
 - Modeling window
 - Model tab, 298
 - Tool Manager turn tool, 338
- Toshiba control - see "Tosnuc", 528
- Tosnuc control, 528
- Transformation Matrix - Surface to Solid window, 245
- Translate, 299
 - component or model, 95
 - coordinate system, 130
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - tool holder, 347
 - Tool Manager "Position" window, 347
 - tool path, 130
- Travel limit errors - see "Overtravel errors", 194
- Travel Limits - Machine Settings window, 387
- Triangle - Tool Manager turn tool, 338
- Troubleshooting - see "Errors", 674
- truecolor - command line option, 467
- Turn tools, 157
 - about describing, 157
 - control point, 157
 - describing in Tool manager, 338
 - turn tool components, 157
- Turning, 315
 - APT Settings window tab, 331
 - debug messages for, 327

Help on VERICUT

- defining cutting tools for - via Tool Manager, 338
- G-Code Settings window tab, 318
- GOTO Format - APT setting, 331
- machining type, 315
- merged turning, 318
- setting Default Machine Mode for, 362
- TURRET record, 508
- Type, 352
 - Type - Add/Modify Word window, 352
 - Type - G-Code Settings window
 - Variables tab, 318
 - Type - Modeling window
 - Model tab, 298
 - Type - ShadeCopy window
 - Attributes tab, 268
 - Type - Tool Manager "Position" window, 347
 - type C - cutter compensation, 420
 - Type II Syntax - Add/Modify Word windows, 352
 - Type of Subroutine Names - Subroutines control settings window, 377
- U Linear - component type, 307
- UG Interface - license, 19
- Undo, 347
 - Coordinate System window, 336
 - Modeling window
 - Assemble tab, 301
 - Position tab, 299
 - Tool Manager "Position" window, 347
- Unit Conversion - in file selection window, 233
- Unit Power - OptiPath Manager window, 402
- Units, 34
 - AUTO-DIFF window
 - Boundary tab, 394
 - changing for a new session, 44
 - default tool path units, 331
 - default units for pre-4.0 VERICUT files, 261
 - Export Model window, 252
 - Modeling window
 - Model tab, 298
 - new session units, 44
 - session units, 34
 - Tool Add/Modify/Copy windows, 338
 - UNITS record, 508
 - Units to Save - in Control file, 351
 - Units to Save - in Machine file, 386
 - Units to Save - in User file, 235
- unsigned incremental circle motion, 365
- Use Holders - setup Toolpath window, 311
- Use Tool List - setup Toolpath window, 311
- User Defined, 380
 - Setup > Control > Settings function, 380
 - Setup menu > G-Code > Process Options function, 328
- User file, 42, 481
 - command line option for, 467
 - description of, 481
 - opening, 46
 - saving, 46
- User interface, 34, 479
 - BUSY-OptiPath lights, 41
 - changing font size, 479
 - graphics area, 35
 - look and feel, 287
 - message area, 38
 - progress bar, 39
 - simulation (VCR) controls, 40
 - toolbar, 37
 - VERICUT main window, 34
- user variable override, 467
- User variables/values, 42, 233, 467, 481
- uservariablename=<uservalue> - command line option, 467
- Using Shortcut to access files, 42
- Using VERICUT, 31
- usr file - see "User file", 481
- usr= - command line option, 467
- UV Relative to XY - Wire EDM control settings window, 374
- V Linear - component type, 307
- Value - G-Code Settings window
 - Variables tab, 318
- Value Type - Add/Modify Word window, 352
- Values - Tables window feature, 191
- Variable - Add/Modify Word/Address windows, 357
- Variable - see also "Variables", 318
- Variable Name - Add/Modify Event window, 381
- Variable Name - Add/Modify WordAddress window, 357
- Variable/Macroname option - Add/Modify Word/Address windows, 357
- Variable/Word option - Add/Modify Word/Address windows, 357
- Variables, 318
 - adding, 145
 - debug messages for, 327
 - defining tag word for, 352
 - deleting, 145
 - environment variables, 479
 - modifying, 145
 - monitoring, 318
 - seeing current values, 318
 - setting via event, 381
 - setting via word/address, 357
 - Variables - G-Code Settings window tab, 318
- VCR control icons, 40
 - Image file playback, 266

Help on VERICUT

- simulation control, 40
- VDA (VDASF) converter, 247
 - command line options, 474
 - converting VDA data, 107
 - data converted by, 108
 - description of, 107
 - VDA version support, 108
 - window features, 247
- Verband der Automobilindustrie, 247
- VERICUT, 1
 - file find logic, 42
 - functions - listed by menu, 229
 - functions - map of 4.4 functions in 5.0, 222
 - how to access, 14
 - introduction to, 14
 - licenses, 18
 - main window features, 34
 - overview of using, 31
 - resetting the model, 40
 - specifying units for a new session, 44
 - starting a new session, 44
 - starting the simulation, 40
 - stopping the simulation, 40
 - VERICUT files - descriptions of, 481
 - VERICUT initialization files, 481
 - VERICUT Linear-sweep file, 481
 - VERICUT Log file, 481
 - VERICUT model files, 481
 - VERICUT Polygon file, 481
 - VERICUT Sample files, 481
 - VERICUT Solid-of-revolution file, 481
- Verification - license, 19
- View - OptiPath Data Link window, 416
- View APT Output - Setup menu > G-Code > Process Options function, 329
- View coordinate system, 57
- View G-code Log - Setup menu > G-Code > Process Options function, 330
- View List, 283
 - creating, 73
 - managing, 283
- View menu, 229
 - Tool Manager window, 338
 - VERICUT main window, 229
- View Name - for selecting/storing views, 283
- View Type - View Attributes window, 279
- viewport - see "views", 35
- views, 275
 - about, 57
 - adding, 67
 - cascading, 68
 - changing view type, 69
 - deleting, 67
 - deleting stored view, 75
 - fitting, 64
 - in graphics area, 35
 - layout, 66
 - light source, 71
 - modifying stored view, 74
 - panning, 63
 - printing, 186
 - renaming, 74
 - renaming stored view, 74
 - reversing, 61
 - rotating, 59
 - sectioning, 65
 - selecting, 73
 - storing, 73
 - tiling, 68
 - types, 279
 - zooming, 62
- Visibility - Modeling window
 - Component Attributes tab, 304
- V-ISO - View Orient window, 277
- Volume, 391
 - calculate for workpiece and uncut stock, 179
 - Volume Removal - optimization method, 402
 - X-Caliper window tab, 391
- W Linear - component type, 307
- Walls, 279
 - background attribute, 279
 - color assignment, 257
 - displaying, 72
 - setting "Machine Up" direction for, 387
- Warning messages, 674
- Warnings - Status window, 289
- Wear - Advanced OptiPath Settings window, 402
- Welcome to VERICUT, 1
- Width - Tool Manager turn tool, 338
- Width (Y) - Modeling window
 - Model tab, 298
- Width Table - OptiPath Manager window tab, 402
- Windows - Look & Feel option, 287
- Wire Diameter - G-Code Settings window
 - WireEDM tab, 318
- Wire EDM, 318
 - configuring control settings for, 220
 - configuring G-code settings for, 153
 - debug messages for, 327
 - G-Code Settings window tab, 318
 - setting Default Machine Mode, 362
 - Setup > Control > Settings function, 374
- Word - Add/Modify Word/Address windows, 357
- Word Format - Setup menu > Control, 352
- Word functions - see "Functions", 463
- Word list - Find Word/Address window, 357
- Word Name list - Words window, 352
- Word Value Spacing - OptiPath control settings

Help on VERICUT

- window, 378
- Word/Address, 357
 - see also - "Groups", 357
 - Setup menu > Control function, 357
- Word/Variable option - Add/Modify
 - Word/Address windows, 357
- words - interpreted by NC control, 197
- Words/Range - Find Word/Address window, 357
- Work offsets, 435
 - Base Work Offset table, 419
 - setting initial work offset, 376
 - Work Offset table, 419, 435
- Work Table to XY Output Plane - Wire EDM control settings window, 374
- Work Table to XY Plane - G-Code Settings window
 - WireEDM tab, 318
- Working Directory, 239
 - File menu function, 239
 - setting, 45
- WorkNC Interface - license, 19
- Workpiece Origin - View Axes window, 284
- X - Tool Manager mill tool profile, 338
- X Center - Tool Manager mill tool profile, 338
- X Linear - component type, 307
- X+/X- rotation options, 299
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position" window, 347
 - View Orient window, 277
- X-Caliper - Analysis menu, 391
- XY Also Moves UV - Wire EDM control settings window, 374
- XY Plane to UV Plane - G-Code Settings window
 - WireEDM tab, 318
- XY-YX-YZ-ZY-ZX-XZ - View Orient window, 277
- XYZ data fields - X-Caliper window, 391
- Y Linear - component type, 307
- Y+/Y- rotation options, 299
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position" window, 347
 - View Orient window, 277
- Yasnac control, 528
- YZ-ZY-ZX-XZ-XY-YX - View Orient window, 277
- Z - Tool Manager mill tool profile, 338
- Z Center - Tool Manager mill tool profile, 338
- Z Linear - component type, 307
- Z+/Z- rotation options, 299
 - Coordinate System window, 336
 - Modeling window
 - Position tab, 299
 - Tool Manager "Position" window, 347
 - View Orient window, 277
- Zoom, 277
 - via command line option, 467
 - view, 62
 - Zoom - View Orient window feature, 277
 - Zoom In-Out - View Orient window features, 277
- zoom_out - command line option, 467
- ZX-XZ-XY-YX-YZ-ZY - View Orient window, 277