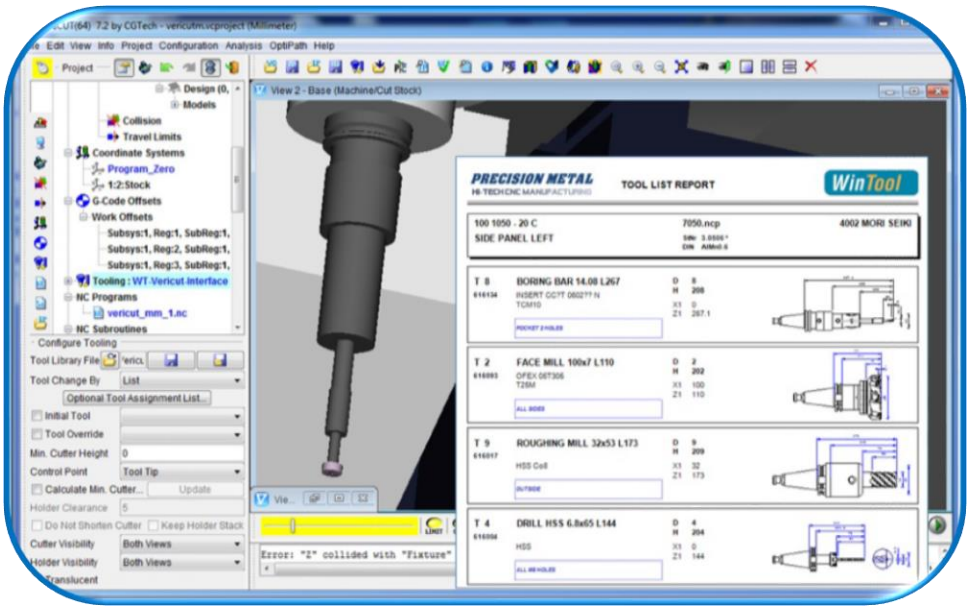
WT-VERICUT Interface



Manual *WinTool* Interface 3.2.0 for VERICUT

The WT-VERICUT-Interface enables the user to transfer assemblies from the *WinTool* database to the VERICUT Tool Manager.

Requirements:

- *WinTool* 2011 Professional or later
- VERICUT 6 or later

WinTool AG
Flüelastrasse 7
CH-8048 Zürich
Phone: +41 (0)44 401 00 55
info@wintool.com
https://www.wintool.com

Contents

Summary 3
Job 3
Requirements 3
Supported tool types 3
Licensing 3
Copyright 3
Installation 4
Directory Structure 4
Interface Installation 4
Update Installation 4
Interface Run 5
Interface Configuration 6
General Information 6
Configuration Window 6
Description 7
Configuration File 8
Integration in VERICUT 8
Working with the WT-VERICUT-Interface 11
User Classification 11
Activation in WinTool 12
Use of WT-VERICUT Interface 12
Updating obsolete *WinTool* Tool Lists in VERICUT 14
Prepare Assembly Data in *WinTool* 15
Standard Assembly 15
User Models 15
Cutter user model support of tool types 16
Create a DXF-contour 17
Annex 18
Troubleshooting 18
Insert type is unknown. Check insert… 18
Insert type is supported using a step file… 18
List of Figures 19

Summary

Job

The WT-VERICUT-Interface enables the user to transfer assemblies from the *WinTool* database to the VERICUT Tool Manager. Full graphic representation of the assembly data in the *WinTool* database is supported which avoids redundant and divergent assembly definitions in programming, NC code verification, and tool setup. The use of the VERICUT interface results in significant time savings during the production preparation and quality assurance in the execution.

Requirements

This interface requires *WinTool* Professional 2011 or later. If *WinTool* 2012 is installed, SP1 is required. *WinTool* must be started to use the Interface.
The VERICUT TLS in the XML format is used by VERICUT 6 or later.

Supported tool types

All rotating tool parts like holders, extensions, drills, taps, and mills are supported. The *WinTool Shape* Module is used to generate an assembly contour. This module is further described in the corresponding documentation enclosed in the *WinTool* software.

Turning tools with inserts or rotating cutters are also supported. More specifically for holder and insert parts a STEP file is generated and used into VERICUT and for insert parts only, if there is no STEP file available into WinTool then inserts being created using the parametric implementation. Special tools and turning tools, which are not supported by the shape module, can be manually defined as a VERICUT TLS file.

Licensing

You must obtain a signed license agreement from *WinTool* AG in order to be allowed to use the interface.

Copyright

This documentation as well as the WT-VERICUT-Interface software is under copyright of

WinTool AG
Flüelastrasse 7
CH-8048 Zürich
Phone: +41 (0)44 401 00 55
info@wintool.com
https://www.wintool.com

Installation

Directory Structure

WT-VERICUT-Interface 2.9 introduces a clear separation of program files and user data.
All user data is centrally placed the [Public Documents]\WT-VERICUT-Interface folder:

User data	New location
Default location of UserModels folder	[Public Documents]\WT-VERICUT-Interface\UserModels
Default location of Exchange folder	[Public Documents]\WT-VERICUT-Interface \Exchange
Configuration file: WT-VERICUT-Interface.cfg	[Public Documents]\WT-VERICUT-Interface

Note: [Public Documents] on Windows 10/11 is located in C:\Users\Public\Documents\

Interface Installation

Install *WinTool* Professional first, before installation of the WT-VERICUT-Interface.

WinTool must be started up at least once before the WT-VERICUT-Interface can be installed.

Start setup.exe

Update Installation

Start setup.exe and install the interface.

If you are updating from version 2.8 or older:

- If the interface was installed in the **same directory** as the previously installed version, the configuration files are automatically moved to [Public Documents]\WT-VERICUT-Interface.
- If you chose a **different directory** for the interface, you must copy the configuration files manually from the previous installation directory to the directory [Public Documents]\WT-VERICUT-Interface
- The default location of the setting UserModelPath has changed. If you haven't set a UserModelPath in the interface configuration, in which case the UserModels folder is in the interface installation directory, you must move the contents of the folder to the new default location [Public Documents]\WT-VERICUT-Interface\UserModels.

Check the interface configuration using the configuration window (*See configuration*).

Interface Run

WT-VERICUT-Interface start-up button can be found into Tool Manager of VERICUT, Import Tab, Partners Panel in VERICUT 8.2 or newer version.

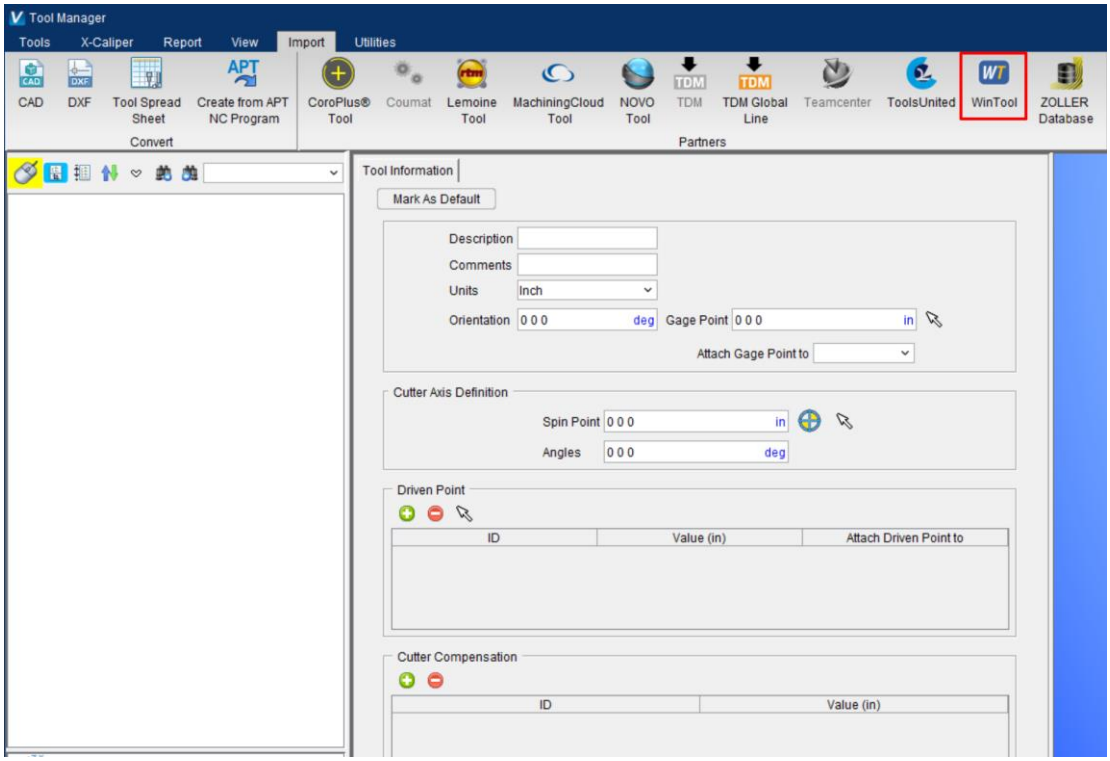


Figure 1 WinTool Button at VERICUT Tool Manager -> Import Tab -> Partners Panel

Interface Configuration

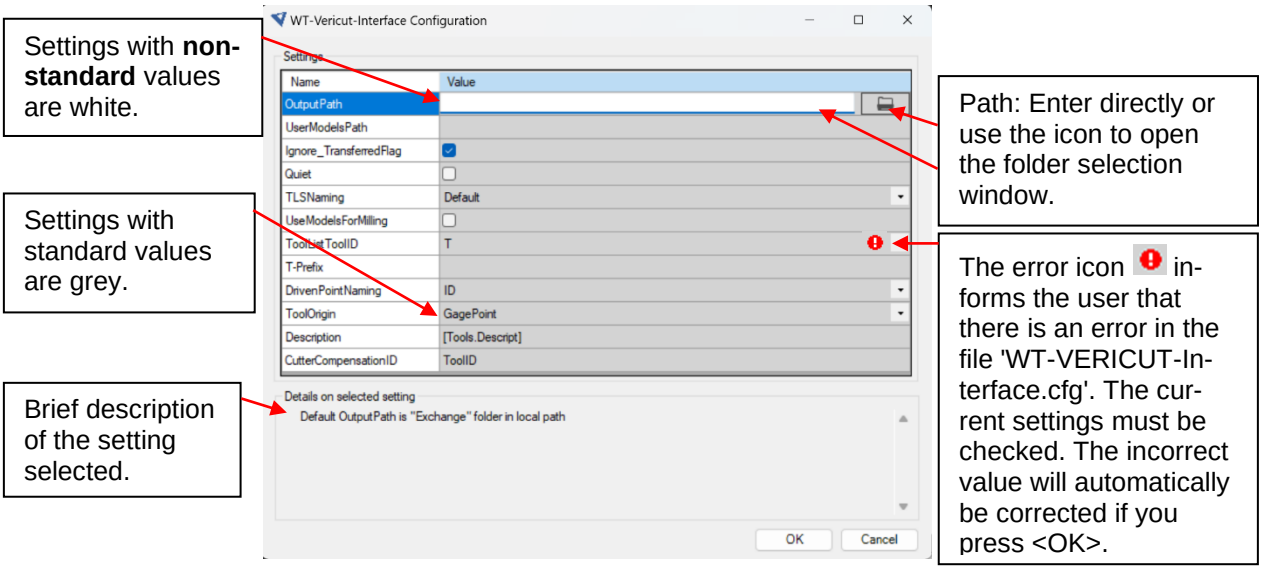
General Information

All configurable settings will be installed with default values. They can be changed in the configuration window or directly in the cfg files (configuration file) if necessary.

Open the configuration window in START > All Programs > WinTool > WT-VERICUT-Interface > WT-VERICUT-Configuration

Configuration Window

The configuration window allows you to check and change the settings of the WT-VERICUT-Interface



<OK> stores all settings. <Cancel> exits the configuration window without saving.

Figure 2 Instruction for the WT-Vericut-Interface

`OutputPath`	Directory path in which the WT-VERICUT-Interface stores the automatically produced tool data exchange file (TLS). Default setting for the OutputPath is the folder "Exchange" in the `[Public Documents]` directory.
`UserModelsPath`	Directory path in which the user models (TLS, DXF, STL, PLY, STEP) are stored. Default UserModelsPath is the folder "UserModels" in the `[Public Documents]` directory.
`Ignore_TransferredFlag`	Must be set to **`True`**. If true, the .dxf files in the UserModels folder are generated by the Shape Module with each transfer (unless the UserModels flag of the tool assembly is activated see chapter below).
`Quiet`	If `True`, the WT-VERICUT-Interface doesn't show any error messages during the transfer.
`TLSNaming`	Defines the name of the generated .tls file when a "tool list" is imported. The file name for a "single tool export" or a "machine type tool export" is always "WT-VERICUT-Interface.tls" It can either be the default (WT-Vericut-Interface.tls), The tool list's "NC-Program", "File Name", "List Name", "Desription" or "O-Number" field.

`UseModelsForMilling`	If `UseModelsForMilling` is set to **`False`**, the assigned 3D Models on milling tools in WinTool, will be exported and used into VERICUT tool assembly. If `UseModelsForMilling` is set to **`True`**, then a contour is used to represent the assembly and the speed of import to VERICUT is quite faster.
`ToolListToolID`	When importing a tool list, this setting determines if the tool ID in VERICUT is transferred from the *WinTool* Ident No., the T-No or the alt. code.
`T-Prefix`	Defines the prefix that is set in front of the tool ID in VERICUT.
`DrivenPointNaming`	Default naming of driven points. Possible values are ID = Tool ID as per parameter “TollListToolID” One = Number “1” None = do not create driven points
`Description`	Description of the assembly as imported into Vericut.

Description

As in Vericut the Description is used to better identify tools, the interface makes it configurable how the Description is generated while importing tool assemblies from WinTool, making imports more flexible. It is up to the user to define a custom naming convention. The interface, however, will provide a default convention which is backward compatible with older Vericut Interface installations. With the new Vericut Interface 2.13 (and newer) placeholders (put in square brackets) can be used to modify the Description. Most Tool values made in WinTool are supported. A short list of supported placeholders:

> *Tools.Nr, Tools.TNumber, Tools.Comment, Tools.Name, Tools.MachineNr, Tools.Descript, Tools.Design, Tools.MaskNr, Tools.ToolWidth, Tools.ToolLength, Tools.OldName, Tools.MDate, Tools.StockState*

Special placeholders with dependent meanings:

- *TNumber* – (without Tools prefix) will become T from Lists if a list is imported, otherwise T from Tools if a tool is imported.

Important Notes:

- Placeholders have to be put in square brackets.
- Description length may be limited in Vericut.

Example:
A setting like

> [Tools.Nr] - [TNumber] - [Tools.Descript]

could be translated to

> 616021 - 0 - End Mill HSS 4x19 4FL

if imported via Tool assembly, or to

> 616021 - 123 - End Mill HSS 4x19 4FL

when imported via Tool list.

`ToolOrigin`	Default value is “GagePoint”. If chosen “DrivenPoint” it will take the value of the Gage Point and negate it.
`CutterCompensationID`	Default value is “ToolID”. If changed it will take the value written in it.

Configuration File

The configuration window reads and stores settings in the file WT-VERICUT-Interface.cfg in the directory [Public Documents]\WT-VERICUT-Interface
The file can be edited with a text editor (e.g. Windows Notepad). The lines starting with a “#” symbol will be ignored by the interface software (the symbol “#” defines a “comment line”). If you remove the # symbol the line will be activated.

Integration in VERICUT

R-click on the VERICUT desktop icon and select "Open file location".

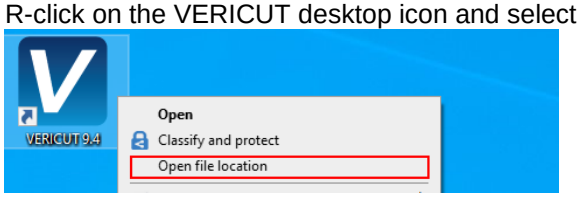


Figure 2 Open VERICUT file location

Copy the directory from the address bar of the folder that appeared.

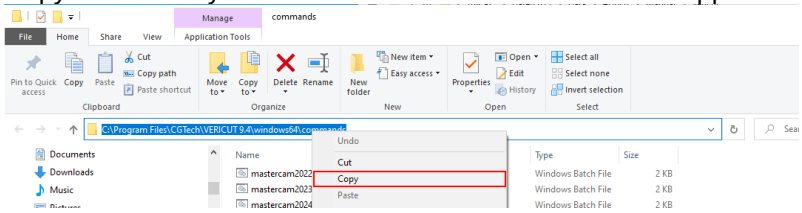


Figure 3 Copy file location directory

Search on Windows for “Notepad” and click on “Run as administrator”.

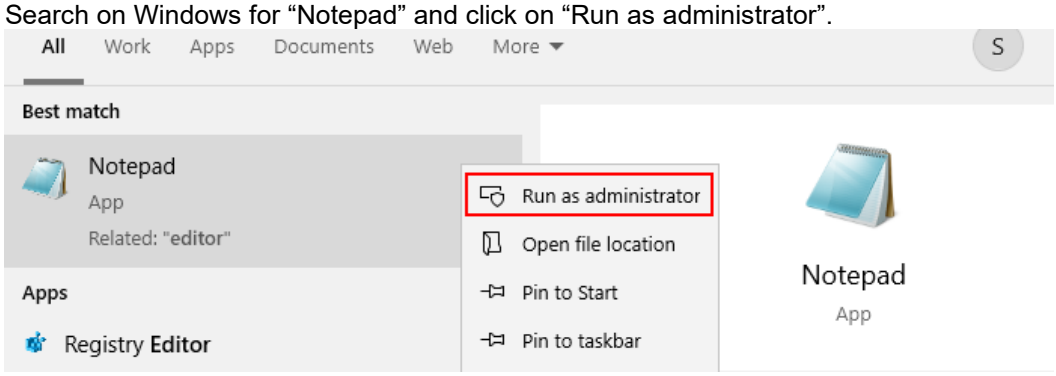


Figure 4 Run Notepad as administrator

In Notepad application click File and then Open. Paste the directory in the address bar, change the filter to “All files”, select vericut.bat and click Open.

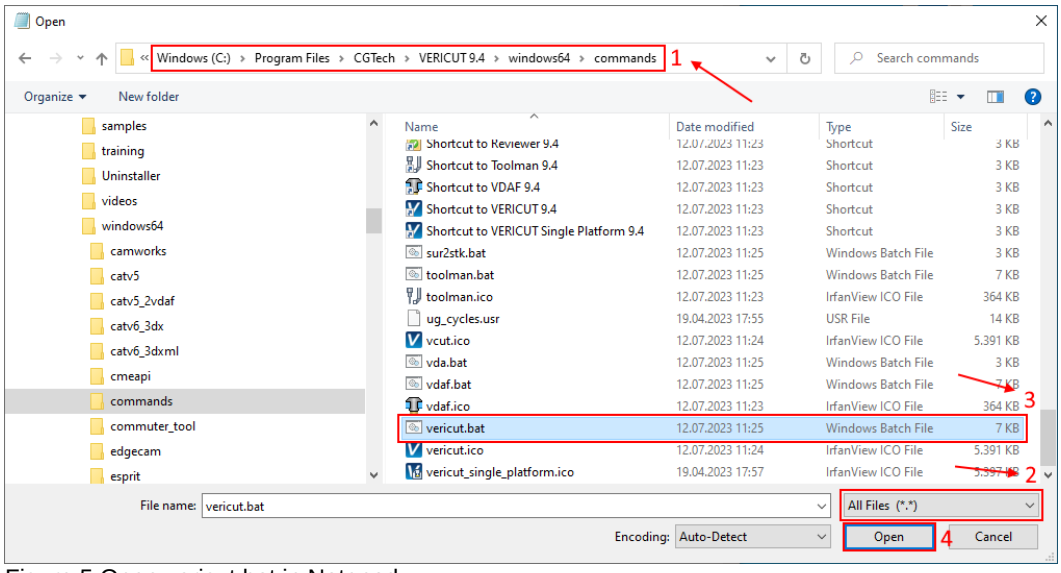


Figure 5 Open vericut.bat in Notepad

Then add the `OutputPath` *(See configuration)* directory of the WT-VERICUT-Interface configuration in the vericut.bat file as shown below and click File and Save.

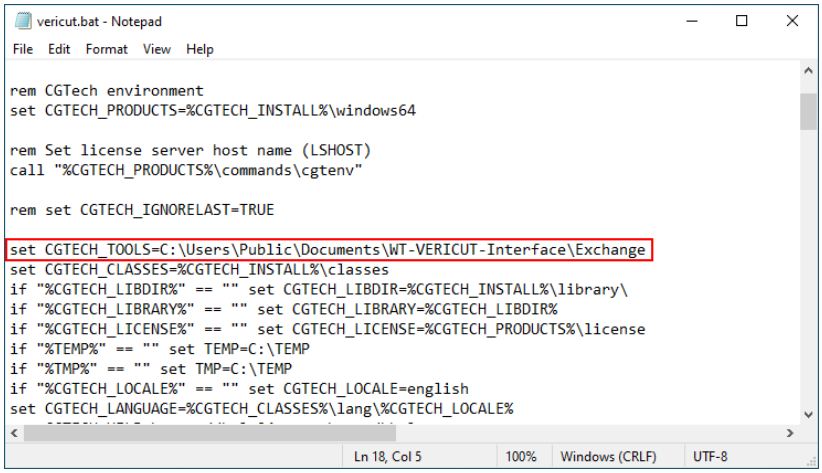


Figure 6 Add VERICUT output path to vericut.bat

R-click on “Tooling” in Project Tree and click Open. Select “CGTECH_TOOLS” as Shortcut.

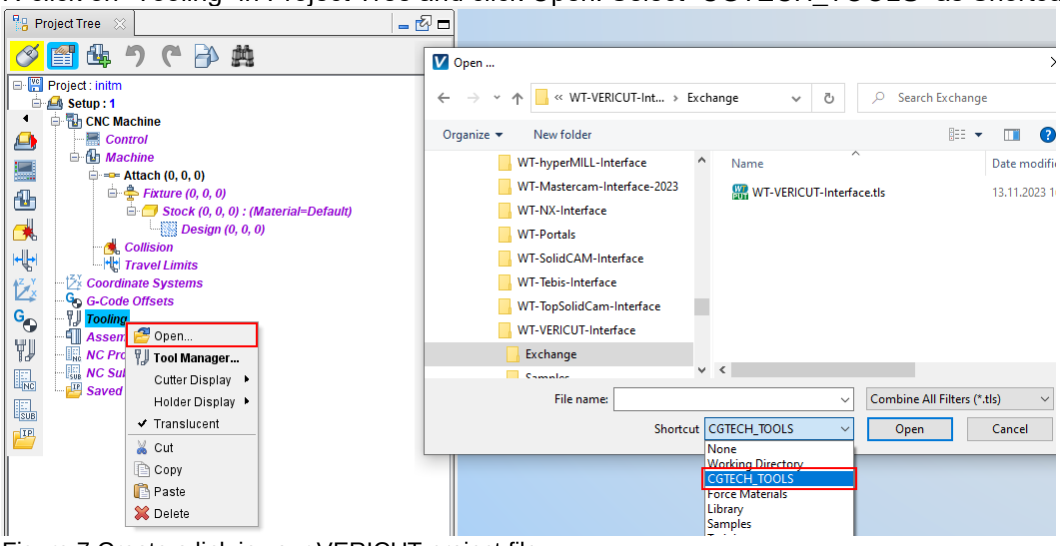


Figure 7 Create a link in your VERICUT project file

Now you can easily find the Interface-created TLS files in your current VERICUT project using the "Shortcut" combo box and select "WT-VERICUT-Interface.tls" to link it to your VERICUT project file:

If you make changes to the tool list in *WinTool*, you can update the VERICUT project easily:

- Create a new WT-VERICUT-Interface.tls using the WT-VERICUT-Interface
- Update your VERICUT project files using the following function:

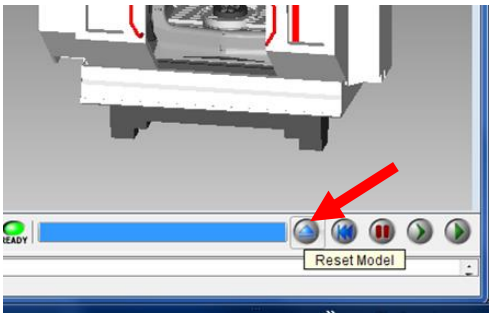


Figure 8 Reset Model to update your project

Working with the WT-VERICUT-Interface

User Classification

Add the corresponding VERICUT tool type in the data field "Note" of each *WinTool* User-Classification. You can configure this in the menu "Settings" in "Classes".

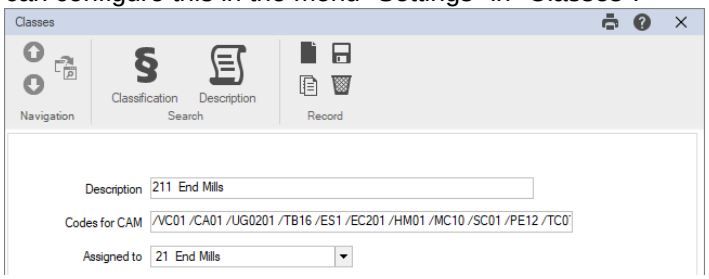


Figure 9 Configurate the tool classes

VERICUT tooltype name		*WinTool* tooltype	Used for
Ignore (Special type)		/VC00	Measurement equipment etc., see note below
	Milling	/VC01	Milling- drilling- boring tools, tap tools (tool profile), reamers
	Turning	/VC02	Turning, grooving and threading tools
	Tap	/VC03	Tap tools (parametric)
	Probe	/VC04	Probe (parametric)

Note: If tool type "`Ignore`" is assigned to a user classification, the tool assemblies assigned to this class will be excluded from the data transfer. This is useful for filtering out measuring equipment etc.
Previous versions of the WT-VERICUT-Interface used the old VERICUT tool types. These will be mapped automatically to the new tool types:

Old	New
/VCFLAT_END	/VC01
/VCBALL_END	/VC01
/VCBULL_NOSE	/VC01
/VCDRILL	/VC01
/VCTRIANGLE_INSERT	/VC02
/VCDIAMOND_INSERT	/VC02
/VCGROOVE_INSERT	/VC02

If no VERICUT tool type has been assigned to a *WinTool* classification, yet, and the setting `Quiet=False` (*See configuration*) has been configured, the user will be asked to assign the VERICUT tool type during the data transfer. Tool assemblies using a *WinTool* classification which has not been assigned to a VERICUT tool type, will not be transferred.

Activation in WinTool

In the *WinTool* Tool Assembly window in the folder tab CAM, you must display the row "Vericut". You can configure this in the menu "Settings" in "CAM settings".

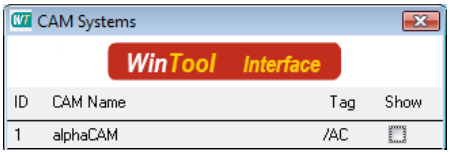


Figure 10 Activate VERICUT Interface in WinTool

Use of WT-VERICUT Interface

In the installation directory start "WT-ToolExport.exe" and select a tool assembly, a tool list, or all tools for a machine type.

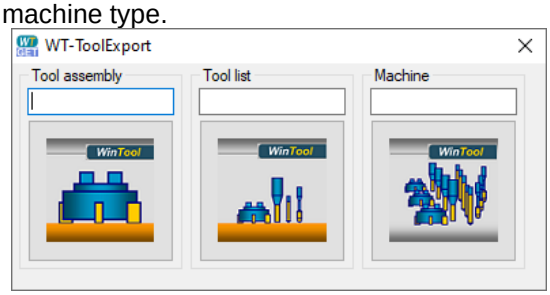


Figure 11 Menu WT-ToolExport

Tip: Create a shortcut on the desktop and pin it to the Windows taskbar.

If no valid user classification is set for a tool assembly and the configuration is set to `Quiet=False` (*See configuration*) you will be asked to assign a tool type.

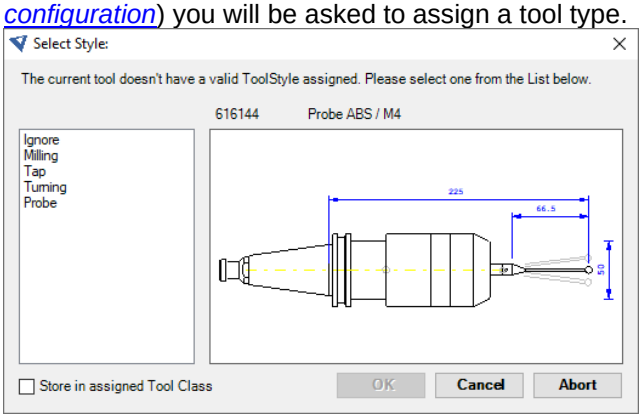


Figure 12 Assign a tool type

If you select "Ignore" to assign to a tool classification, the tool assemblies in this classification will not be transferred at all. This is useful for data that must not be transferred to VERICUT, e.g. measuring equipment. Check “Store in assigned Tool Class” to save the selection in *WinTool* (See "User Classification" above). Click “OK” to confirm, "Cancel" to abort transfer of current tool assembly, "Abort" to abort the complete transfer.

The WT-VERICUT-Interface will create the exchange file in the directory `OutputPath` (*See configuration*).

Start the VERICUT tool manager and open this file to see the tool assemblies:

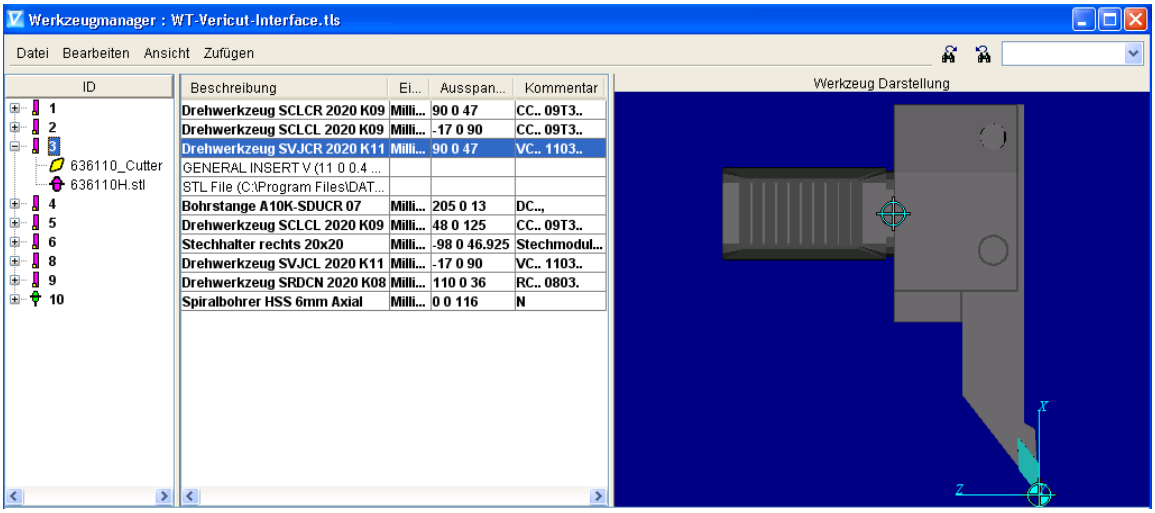


Figure 13 WT- Vericut Interface-overview about the tool assemblies

Note: Milling and tap tools are imported with a default driven point of (0 0 0) and a cutter compensation value of "tool assembly diameter / 2"

Updating obsolete *WinTool* Tool Lists in VERICUT

After a *WinTool* tool list has been imported to a VERICUT project, you can easily update an "obsolete tool list" with a "new tool list": First get the new tool list from *WinTool* using the WT-VERICUT-Interface (this will overwrite a previously created tool list in the `OutputPath` (*See configuration*) directory) and then click in VERICUT the Reset Model button to update the project data:

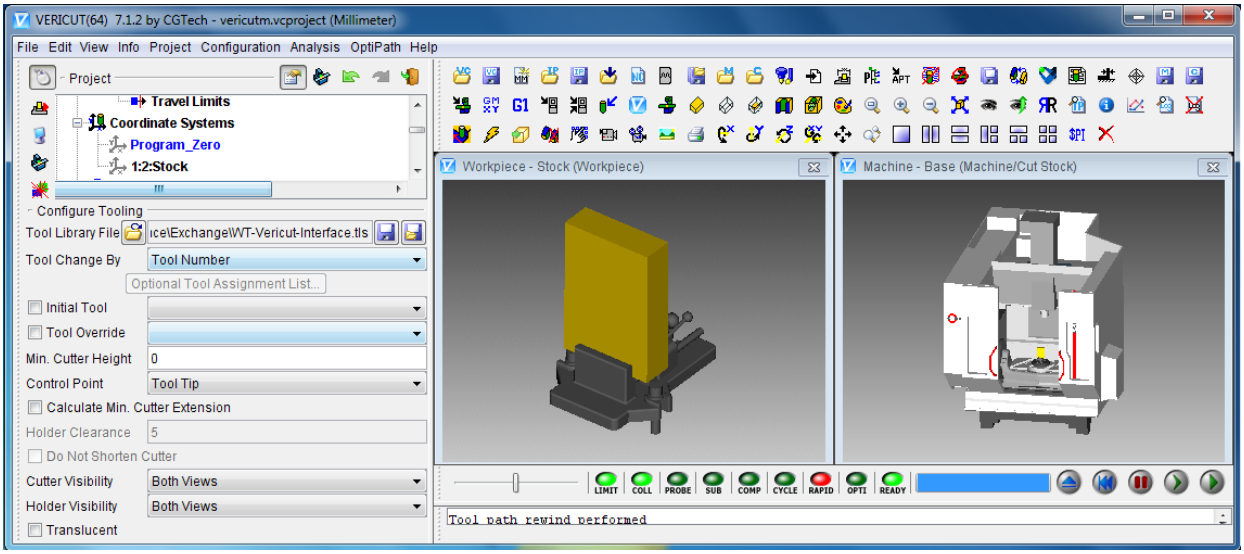


Figure 14 VERICUT 7.1.12 - Reset the Model to update datas

Prepare Assembly Data in *WinTool*

Standard Assembly

Note, that only classified assemblies are being transferred properly. Make sure that:

- Each *WinTool* classification is assigned to the corresponding VERICUT tool type.
- Each tool assembly has assigned a *WinTool* Machine Type.
- Each tool assembly has a "namegiving" and a "cutting" component.

The *WinTool* tool component and *WinTool* assemblies must be recorded correctly according to the specifications (see *WinTool* manuals). Otherwise the data transfer might fail.

User Models

If the Shape module does not support a tool assembly and cannot produce a tool contour, a user model can be used instead. WT-VERICUT-Interface uses the following rules for the tool data import of user models:

Using DXF-contours

If the flag UserModel in WinTool is set, the WT-VERICUT-Interface searches in the UserModel directory a *.dxf contour file with the name Tools.Nr (=ID of active tool, e.g. 616142). If it is available, the WT-VERICUT Interface creates a 3D model from this contour. → See chapter "Create a DXF-contour" on page 14.

Using TLS for tool assembly

If in the UserModel directory a *.tls file with the name Tools.Nr is available, a DXF contour will be ignored and a reference will be created (in the tag <ReferenceTool>). All other fields in the header, as well as Cutter and Holder of the tool will be ignored.

Important: The ID in the <Tool> tag of the *.tls file must match with the Tools.Nr of the tool assembly:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<CGTechToolLibrary version="6.1">
  <Tools>
    <Tool ID="616073" Units="Millimeter">
      <Description>twist drill HSS 2.5mm</Description>
      <Comments>N</Comments>
      <Teeth>2</Teeth>
      <Type>Milling</Type>
      <Cutter FluteLength="30">
        <SOR ID="616073_Cutter">
          <Pt>
```

→ See sample "616073.tls" in the "Samples" folder in the [Public Documents]\WT-VERICUT-Interface directory.

Using TLS for tool assembly holder and cutting part
If in the UserModel directory a *.tls file with the name Tools.Nr +"H" (Holder) is available, the "partial tool model" will replace the holder portion of the DXF contour model and a reference will be created (in the tag <Reference>).

If in the UserModel directory a *.tls file with the name Tools.Nr +"C" (Cutter) is available, the "partial tool model" will replace the shank and the cutting portion of the DXF contour model and a reference will be created (in the tag <Reference>).

Important: The ID in the <Tool> tag of the *.tls file must match with the Tools.Nr of the tool assembly. Additionally, the ID in the <SOR> tag inside the <Holder> or <Cutter> tag must match Tools.Nr + "_Holder" or Tools.Nr + "_Cutter", respectively:

```
<?xml version="1.0"?>
<CGTechToolLibrary Version="7.0">
  <Tools>
    <Tool ID="616073" Units="Millimeter">
      <Holder>
        <SOR ID="616073_Holder">
          <Pt>
```

→ See sample "616073H.tls" in the "Samples" folder in the [Public Documents]\WT-VERICUT-Interface directory.

Using 3D model for tool assembly holder and cutting part
If in the UserModel directory a 3D model file with the name Tools.Nr or Tools.Nr +"H" (Holder) is available, the "partial tool model" will replace the holder portion of the DXF contour model. A link to the file is created (in the tag <Model>).

If in the UserModel directory a 3D model file with the name Tools.Nr +"C" (Cutter) is available, the "partial tool model" will replace the shank and the cutting portion of the DXF contour model. A link to the file is created (in the tag <Model>).

Supported model types are: STL, STK, PLY and STEP.

See section "**Error! Reference source not found.**" on page **Error! Bookmark not defined.** for instructions on how to create a 3D model.

Note:
Turning tool holders need a 3D model in order to be transferred in VERICUT.

The models are transferred with rotation and origin parameters according to the STL's supplied with WinTool. The parameters can be changed by modifying the setting `UseModelParameters` (*See configuration*).

Cutter user model support of tool types

Tooltype		3D model	TLS file	DXF CUT contour
	Milling	Yes	Yes	Yes
	Turning	Yes	Yes	No
	Tap	No	Yes	No

Note: All user model types are supported for holders.

Create a DXF-contour

For custom assemblies you can create the DXF file manually.

Use the *WinTool* Shape module to create a contour. Even if a tool is not supported fully by the Shape Module, it will create in most cases a contour-DXF, although not with all additional details of the custom tool - but with a lot of useful elements in place already: holder, extensions, reductions, shank, total length, correct layers, etc.

Then modify it with *Vector* until it is exact. You must use the layers CUT, NOCUT, and SHANK:

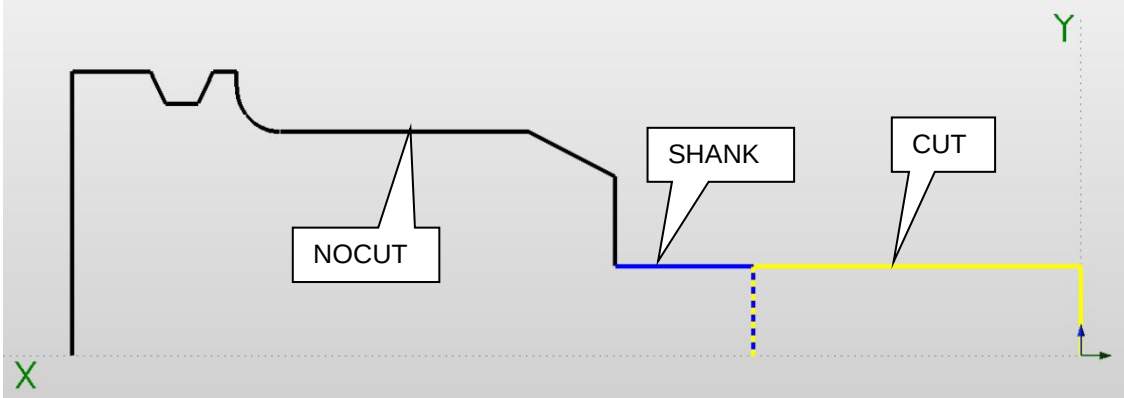


Figure 15 Manual tool contour DXF

The CUT layer is independent from the shank and holder closed contour. It **must** start and end at the X-axis (Y=0). Only the first and the last line of the contour are allowed to start/end at X-axis.

The SHANK and NOCUT layer together must form a closed contour. It **must** start and end at the X-axis (Y=0). Only the first and the last line of the contour are allowed to start/end at X-axis.

When you have finished the modification, you must save the file in the UserModel directory with the name of the *WinTool* tool assembly Ident No (e.g. `616089.dxf`)

Assign the new DXF to the *WinTool* tool assembly: Check the box “User Model” in the tab “CAM” in the row containing "Vericut". If it is missing, please activate VERICUT in "Settings" > "CAM settings" on the main *WinTool* screen.

DXF contour with non-cutting section in cutter
For tool assemblies which have non cutting sections (e.g. face mills, tap tools...) the SHANK must extend into the CUT lines to create the non-cutting section:

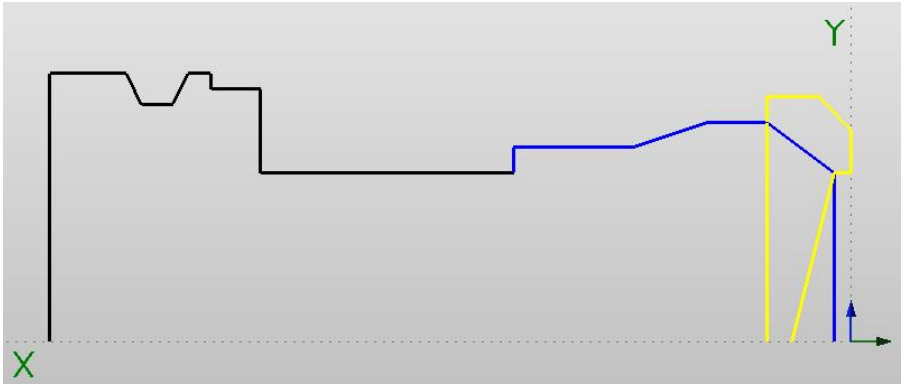


Figure 16 Contour with non-cutting section

Annex

Troubleshooting

Insert type is unknown. Check insert…

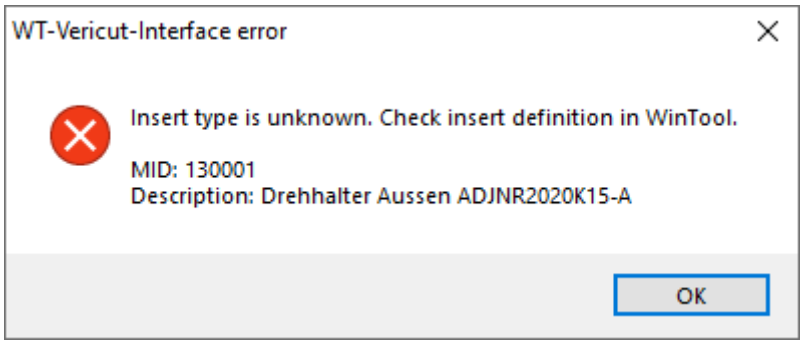


Figure 7 Error message: Insert type is unknown. Check...

Cause: The type of the used insert cannot be determined by the WT-VERICUT-Interface or the data (component) might be incorrect.

Solution: Check the component data for accuracy and correct if necessary.

Insert type is supported using a step file…

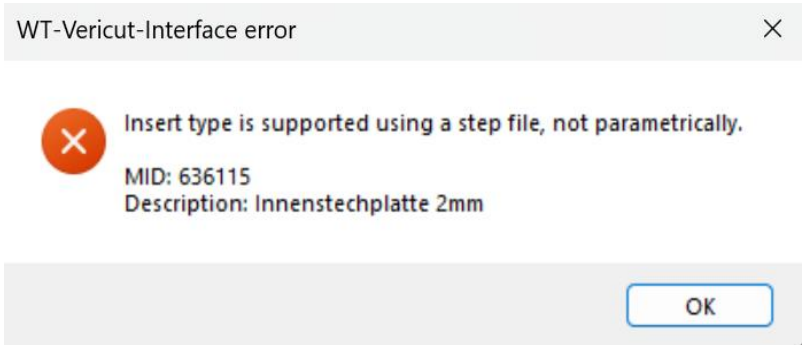


Figure 18 Error message: Insert type is supported using a step file...

Cause: Vericut does not support this type of insert parametrically and can be imported only if there is a step file for its component available.

Solution: Adress a step file for this insert on WinTool.

List of Figures

Figure 1 WinTool Button at VERICUT Tool Manager -> Import Tab -> Partners Panel 5
Figure 2 Open VERICUT file location 8
Figure 3 Copy file location directory 8
Figure 4 Run Notepad as administrator 8
Figure 5 Open vericut.bat in Notepad 9
Figure 6 Add VERICUT output path to vericut.bat 9
Figure 7 Create a link in your VERICUT project file 9
Figure 8 Reset Model to update your project 10
Figure 9 Configurate the tool classes 11
Figure 10 Activate VERICUT Interface in WinTool 12
Figure 11 Menu WT-ToolExport 12
Figure 12 Assign a tool type 12
Figure 13 WT- Vericut Interface-overview about the tool assemblies 13
Figure 14 VERICUT 7.1.12 - Reset the Model to update datas 14
Figure 15 Manual tool contour DXF 17
Figure 16 Contour with non-cutting section 17
Figure 19 Error message: Insert type is unknown. Check... 18
Figure 20 Error message: Insert type is supported using a step file... 18