WT-CAM-InterfaceApp

Manual

WT-CAM-InterfaceApp 1.6

WT-CAM-InterfaceApp writes data from *WinTool* tool assemblies, cutting conditions and tool lists into a XML file. This format allows further processing through CAM specific software which then can transfer tool assemblies into a CAM system.

Requirements

- Min. *WinTool* 2011 Professional

WinTool AG
Flüelastrasse 7
CH-8048 Zurich
Phone: +41 (0)44 401 00 55
http://www.wintool.com

Contents

Job 3
Important remarks 3
In general 3
XML 3
WT-DXF2GEO 3
Development installation 3
Changes in this version 3
XML Output 3
Configuration 4
Configuring WT-CAM-InterfaceApp.exe 4
Configuring WT-CAM-InterfaceApp-ToolTypes.xml 4
WinTool tool classification 5
Assignment window 5
Interface parameters 6
Configuration file 6
Configuration window 8
Start parameters of WT-CAM-InterfaceApp.exe 9
Start of WT-CAM-InterfaceApp.exe via WT-ToolExport 10
Configuration and Parameters 10
Exit codes 10
Output format of WT-CAM-InterfaceApp 10
Datafields 10
XML structure 11
WT-DXF2GEO 13
Create a tool list in WinTool: WT-MakeList 13
History 14
WT-CAM-InterfaceApp 14
WT-ToolExport 14

Job

WT-CAM-InterfaceApp writes data from *WinTool* tool assemblies, cutting conditions and tool lists into a XML file. This format allows further processing through CAM specific software which then can transfer tool assemblies into a CAM system.

Important remarks

In general

Important: If you use the interface for the first time, request a *WinTool* CAM-ID and a CAM tag from WinTool. These IDs will uniquely identify your interface and must be entered in the file "WT-CAM-InterfaceApp-ToolTypes.xml"
Every *WinTool* CAM interface must conform to the guidelines specified in the document "WT-CAM-Interface-Guidelines-E". This ensures a unified structure among all interfaces.

Users of the previous versions should read chapter "Changes in this version".

XML

First time users: The configuration setting `xmlDecSepIsPoint=yes` must be set, see page 7 for details.
See chapter "Changes in this version" below for a detailed change description.

WT-DXF2GEO

Only rotationally symmetric tool assemblies are supported. This is a limitation of the WT-Shape module.

Development installation

Unzip all files in the zip file into a folder.

Changes in this version

XML Output Version 1.6

Change	Element	Datafield Name	Definition
New	<ToolAssembly>		Defaults to the field “Description”, can be configured freely by the user including database contents.

Configuration

Configuring WT-CAM-InterfaceApp.exe

The file WT-CAM InterfaceApp.exe can be renamed, e.g. to WinTool-newCAM-Interface.exe. This name is important because it is used as a reference name in the following configuration files.
It will be referred to as `[Interface-name]` throughout this document.

Configuring WT-CAM-InterfaceApp-ToolTypes.xml

Important: This XML must not be edited by the end user. It is recommended to enable the "read-only" flag on the file.

The name of this XML file must be `[Interface-name]`"-ToolTypes".

Example: If the exe is named "WinTool-newCAM-Interface.exe", the XML file has to be named "WinTool-newCAM-Interface**-ToolTypes**.xml".

The root element in the XML file must also match the exe name.

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<WinTool-newCAM-Interface name="newCAM" tag="NC" camId="103">
  <ToolTypes>
    <ToolType id="212" name="end mill roughing" />
    <ToolType id="113" name="center drill" />
    <ToolType id="00" name="screwdriver" />
     … … … …
  </ToolTypes>
</WinTool-newCAM-Interface>
```

Attributes in the root element:

`name:` Is displayed in *WinTool* in the CAM settings and the tool assemblies in the "CAM" tab.

`tag:` Is used in the WinTool classification as a prefix of the CAM tool type mapping, e.g. /**NC**212 (section "*WinTool* tool classification", see below)
Must be requested at *WinTool* and entered here.

`camId:` Unique, internally used camID. Appears in the *WinTool* CAM settings.
Must be requested at *WinTool* and entered here.

Tooltype in the <ToolTypes> element:
For each CAM tool type, the corresponding `<ToolType>` element must be registered in the <ToolTypes> element. The `id` and `name` attributes must be present.

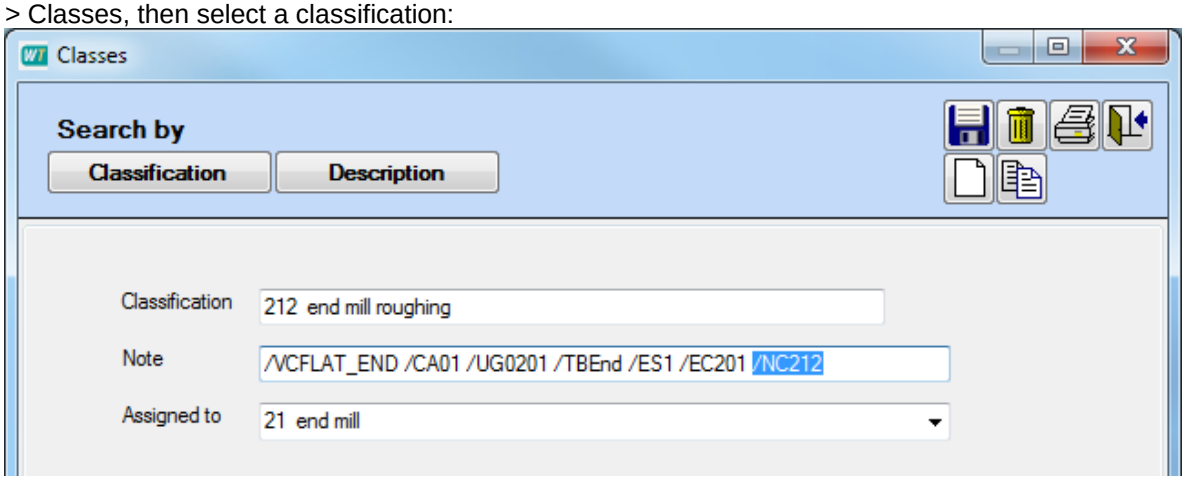
`id:` Is used in the *WinTool* classification as an identifier of the CAM tool type
e.g. /NC**212** (section "*WinTool* tool classification", see below)

`name:` This is the name of the CAM tool type, e.g "center drill". It is displayed if no CAM tool type is assigned to a WinTool classification
(see section "assignment window" below)

WinTool classifications mapped to the `id` "00" are ignored. This means that tools assigned to these classifications are not exported to the XML file.

WinTool tool classification

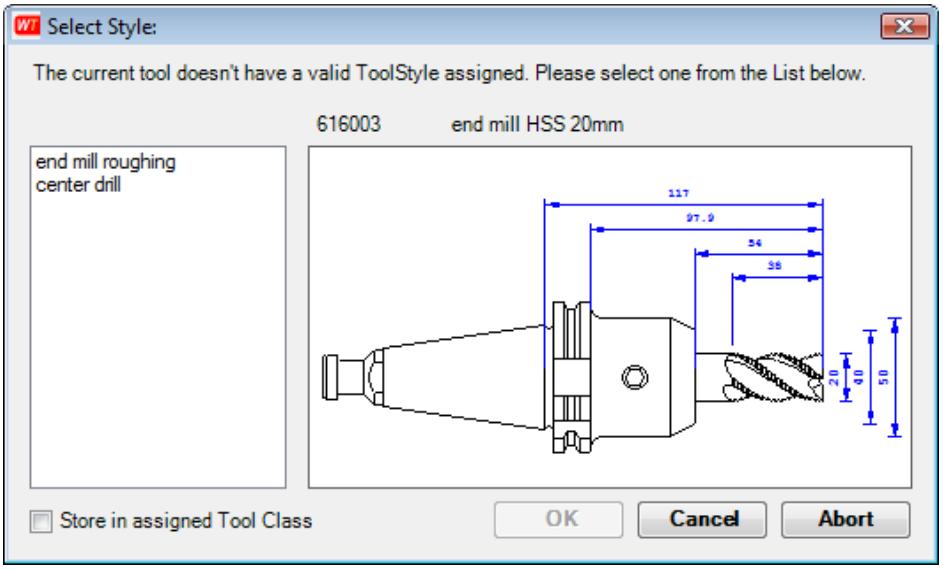
Each *WinTool* tool classification must be mapped to a CAM tool type. If the mapping is missing, the user will be asked (see "assignment window" below) to assign the classification during generation of the XML file. The mapping is stored in the tool classification in the data field "Note". To see it in *WinTool*, select Settings > Classes, then select a classification:



The assignment has the format "/" + `tag` + `id`, e.g. /`NC212`

Assignment window

If the checkbox “Store in assigned Tool Class” is checked, the selected mapping is stored in the *WinTool* tool classification.



Interface parameters

All user data is centrally placed in the directory that is written inside in the text file "AppData". It is located in the installation directory of the WT-CAM-InterfaceApp.
Refer to the document "WT-CAM-Interface-Guidelines-E.pdf", chapter "Installation" for more details.

All configurable parameters are stored in the file `[Interface-name]`**".cfg"** in the `[Directory in file "AppData"]\` directory.
There is also a file `[Interface-name]`**".default.cfg"** in the installation directory of the WT-CAM-InterfaceApp. It contains the default parameter settings.
The parameters can be edited manually in the file or via the configuration window.

Configuration file

In this file, the exe name must be present in the line`[`**`WinTool-newCAM-Interface`**`]` so that the parameters are recognized by the interface.

```
# Configuration file for WT-CAM-InterfaceApp
# ------------------------------------------
# (c) 2014 WinTool AG, Zurich

[WinTool-newCAM-Interface]
OutputPath=
#   Default OutputPath is "Exchange" folder in local path

UserModelsPath=
#   Default UserModelsPath is "UserModels" folder in local path

SelectCutData=False
#   If "False" or empty, all cutting data is transferred.
#   If "True", the interface imports cutting conditions for work materials.
#   a selection window opens if
#   there are multiple or no cutting conditions for the material, or if a
#   single tool assembly is transferred.

#   Do not edit the next line!
# xmlDecSepIsPoint=

Description=[Tools.Descript]
#   Description of the tool as displayed in the CAM
#   See manual for description
# --- End of configuration file ---
```

OutputPath
Sets the folder path in which the generated XML file is stored. Default OutputPath is the "Exchange" folder in the `[Directory in file "AppData"]` directory.

UserModelsPath
Sets the folder path in which the tool contour DXF files are stored. Default UserModelsPath is the "UserModels" folder in the `[Directory in file "AppData"]` directory.

SelectCutData
If SelectCutData is deactivated (**false** = default), all cutting conditions are transferred into the XML file.

If this parameter is active (**true**), cutting conditions for work materials are transferred.
For tool assemblies, tool lists and machine tools the import uses a different cutting condition selection procedure:

Import	Selection procedure
tool assembly data	The cutting condition window opens and all available cutting conditions can be selected.
tool list data	The interface imports all cutting conditions available for one material only. The material is automatically preselected by the interface if the field material in the WinTool tool list has been filled in. Otherwise the cutting condition window of the first tool in the list will be displayed and a cutting condition must be selected manually. The material for this value will be registered by the interface and is used to preselect the cutting condition for all succeeding tools of the list. If a tool has more than one cutting condition for the same material or if no cutting condition exists for the material, the interface requests to select it manually.
machine tools data	The interface imports all cutting conditions available for one material only. The cutting condition window of the first tool of the machine will be displayed and a cutting condition must be selected manually. The material for this value will be registered by the interface and is used to preselect the cutting condition for all succeeding tools of the machine. If a tool has more than one cutting condition for the same material or if no cutting condition exists for the material, the interface requests to select it manually.

Note: The cutting condition window appears only if there is at least one cutting condition.

xmlDecSepIsPoint
We strongly recommend to parse floating point numbers using "." as decimal separator. We will eventually remove this setting and "." will be the standard decimal separator.

By default this setting is disabled. In this case, the decimal separator in floating point numbers depends on the localization settings, e.g. German settings use ',', English and Swiss settings '.'.

If this setting is set to "yes" (**xmlDecSepIsPoint=yes**), all floating point numbers in the output file use the xml standard decimal separator ".".

Note: Users must not change this setting! This setting is not visible in the configuration window.

Description
According tot he configuration an additional field «Configured Description» will be created. 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

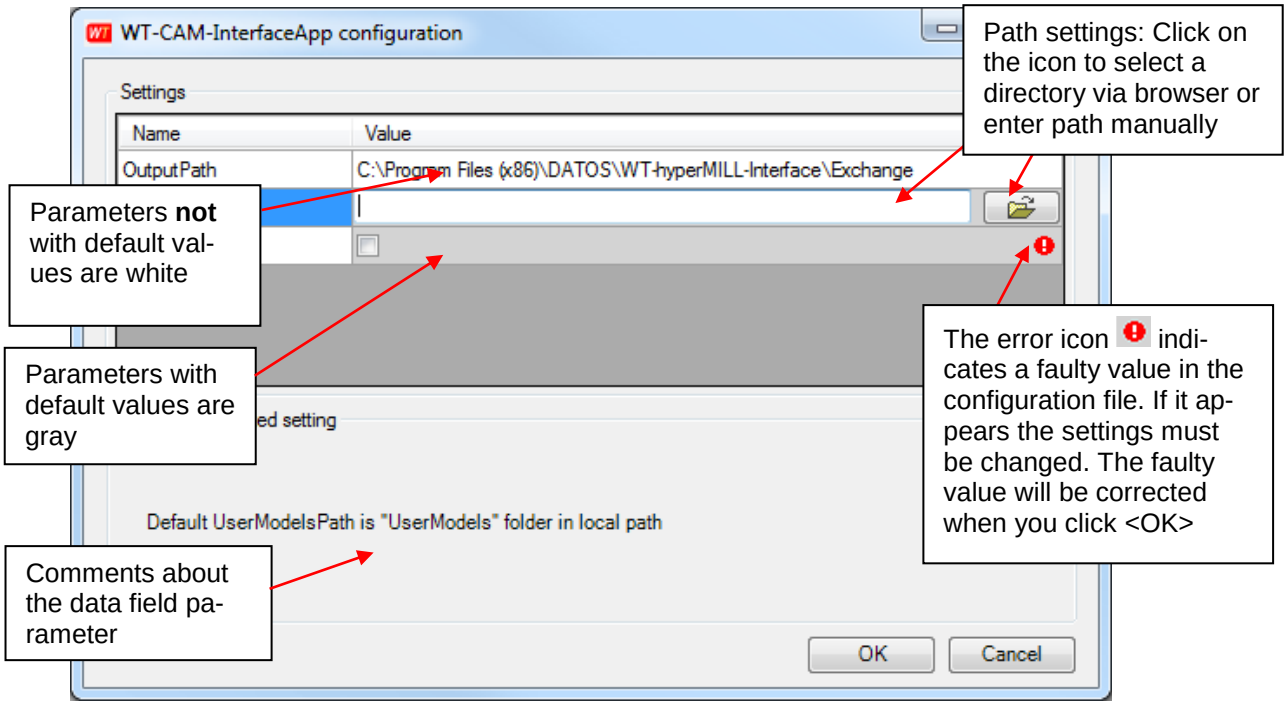
if imported via Tool assembly, or to

616021 - 123 - End Mill HSS 4x19 4FL

when imported via Tool list.

Configuration window

The configuration window is displayed if the WT-CAM-InterfaceApp.exe is started with the parameter "/config".
The configuration file `[Interface-name]`**".default.cfg"** contains the standard configuration values. An error message appears if it doesn't exist.



<OK> stores all parameter values. <Cancel> exits the configuration window without saving.
The configuration window reads and stores settings in the configuration file.

Start parameters of WT-CAM-InterfaceApp.exe

Syntax	Action
/T "Tools.Nr"	Exports tool assembly with nr. "Tools.Nr"
/L "ToolLists.Nr"	Exports all tool assemblies of tool list nr. "ToolLists.Nr"
/I "ToolLists.Ident"	Exports all tool assemblies of tool list with name "ToolLists.Ident"
/M "Machines.Nr"	Exports all tool assemblies attached to machine nr. "Machines.Nr"

Example:
To export tool list nr. 3, the interface must be called like this:
WT-CAM-InterfaceApp.exe /L 3

To export tool list "100 1030 - 10 C", the interface must be called like this:
WT-CAM-InterfaceApp.exe /I "100 1030 - 10 C"

Start of WT-CAM-InterfaceApp.exe via WT-ToolExport

WT-ToolExport.exe is a GUI application which allows the user to search for a tool assembly, tool list or machine in *WinTool*. If the user has selected one of them, it starts the WT-CAM- Interface-App.exe with the according parameters (see next paragraph).

Configuration and Parameters

WT-ToolExport generally doesn't need any configuration file or special command line parameters; however its behaviour can be slightly changed by this.
To be able to use a config file, it has to be named "WT-ToolExport.cfg" and must be placed in the `[Directory in file "AppData"]` directory.
All configuration options can be used thru the config file (separating option / value with "=", or as a command line parameter (separating option / value with " ").

StartMode (T/L/M)

Starts WT-ToolExport just with the given search modes. If e.g. "TL" (without quotes) is used as StartMode, only search for tool assemblies and tool lists is shown. If just one StartMode is given, the search window is directly shown without showing the mode selection window.

PortName ("path-to-file.exe")

Assigns the file to be started after selection. If no path/file name is given, WT-ToolExport will try to find a file according the scheme "WT-*-Interface*.exe" in the path where it is stored.

Exit codes

Type	Exit code
OK	0
User aborted WT-ToolExport	1
Error in WT-ToolExport	2
User aborted WT-CAM-InterfaceApp	101
Error in WT-CAM-InterfaceApp	102

Output format of WT-CAM-InterfaceApp

The output file is in the XML format. The file name is `[Interface-name]`**". xml"**, e.g. "WT-newCAM-Interface.xml". It is stored in the folder **`OutputPath.`**

The tool contour DXF file is stored in the **`UserModelsPath.`** See chapter "4.2.7 CAM" or "4.3.7 CAM" (*WinTool* 2011) in the *WinTool* help to find out more about "Transferred" (<ToolAssembly> "generateShape") and "UserModel" (<ToolAssembly> "UserModel") flag.

Note: With the module "WT-DXF2GEO", a DXF file can be converted into a better processable GEO file. For further information, see the section "WT-DXF2GEO" on page 10.

Datafields

The data fields of the *WinTool* database are stored within the element

```
<DataField name="Nr" value="2" dataType="int" />
```

Possible datatypes:

- int
- double
- string
- bool
- dateTime

XML structure

Export of a tool assembly (/T) and export of all tool assemblies assigned to a machine (/M)

```
<?xml version="1.0" encoding="utf-8"?>
<WT-newCAM-Interface> .........Root element, named after "exe name"
  <ToolAssemblies> ............If >0 tool assembly is imported
    <ToolAssembly>  ...........For each tool assembly, table "Tools".
     <PartNG></PartNG> ........Always present(Namegiving flagged comp.)
     <PartCut></PartCut> ......Always present(Cutting flagged comp.)
     <PartHold></PartHold> ....Optional, usually present
     <PartsArray>  ............Always present
       <Part></Part> ..........For each component, data from
                               tables "ToolParts" and "Parts".
     </PartsArray>
      <CutDataSet>  ...........If >0 cutting condition is present
        <CutData></CutData> ...For each cutting condition, table "CutData"
      </CutDataSet>
    </ToolAssembly>
  </ToolAssemblies>
  <Materials> ..................If >0 material is used in tool list
                                or cutting condition
     <Material></Material>      For each material, data of
                                table "Materials" and "MatClasses"
  </Materials>
</WT-newCAM-Interface>
```

Export of a tool list (/L)

Additional elements are highlighted

```
<?xml version="1.0" encoding="utf-8"?>
<WT-newCAM-Interface> .........Root element, named after "exe name"
 <ToolListData> ..............Always present, data from table "ToolLists"
 </ToolListData> ..............
   <ToolAssemblies> ...........If >0 tool assembly is imported. See table below.
     <ToolAssembly>  ..........For each tool assembly, table "Tools"
      <PartNG></PartNG> .......Always present(Namegiving flagged comp.)
      <PartCut></PartCut>  ....Always present(Cutting flagged comp.)
      <PartHold></PartHold> ...Optional, usually present
      <ToolList></ToolList> ...Always present, data from table "ToolList"
      <PartsArray>  ...........Always present
        <Part></Part> .........For each component, data from
                               tables "ToolParts" and "Parts".
      </PartsArray>
       <CutDataSet>  ..........If >0 cutting condition is present
         <CutData></CutData> ..For each cutting condition, table "CutData"
       </CutDataSet>
     </ToolAssembly>
   </ToolAssemblies>
  <Materials> ..................If >0 material is used in tool list
                                or cutting condition
     <Material></Material>      For each material, data of
                                table "Materials" and "MatClasses"
  </Materials>
</WT-newCAM-Interface>
```

Renamed DB field names in XML

Tools data	
Field name in DB	**Name in <ToolAssembly>**
Classes.Name	ClassesName
T-Number	TNumber
CenterCut	CenterC
CArc	v1
MachineNr	MachNr

Parts data	
Field name in DB	**Name in <Part>**
NameGiving	NG
Classes.Name	ClassName

Cut data	
Field name in DB	**Name in <CutData>**
MatGroups1.Nr	MatGroup1Nr

Special calculations

Tools data	
Name	**Calculations**
SumArcInfluence	Sum of all mounting angle (Bi) values of the tool assembly components **taking into account** the adjustment angles
sumarcinflY	Sum of all mounting angle (Bi) values of the tool assembly components **not taking into account the** adjustment angles
CenterC	• *WinTool* 2014 or newer: If non cutting diam. A4 of namegiving component is 0 `True`, else `False` • Older than *WinTool* 2014: Field "Center Cutting" in *WinTool*

WT-DXF2GEO

The WT-DXF2GEO module converts a *WinTool* tool contour DXF into a GEO file. It is available as an exe file and a dll. It is a C# module.

Note: Only rotationally symmetric tool assemblies are supported. This is a limitation of the WT-Shape module.

WT-DXF2GEO sorts all DXF elements in the DXF file, closes gaps, and creates a GEO file
A GEO file contains a sorted list of GEO elements (lines and arcs) which describe the tool contour, starting with the tip.

The great advantage is that the elements in a GEO file are sorted and can be used directly to draw a 3D model in a CAM system.

More details about GEO and DXF format can be found in section 4.19 or 9.5 (*WinTool* 2011) of the *WinTool* help.

Usage of WT-DXF2GEOApp.exe
WT-DXF2GEOApp.exe /DXF "Path to the DXF file" /GEO "Path to the DXF file"

Example:
WT-DXF2GEOApp.exe /DXF "C:\Interface\UserModels\616001.dxf" /GEO "C:\Interface\UserModels\616001.geo"

Usage of dll
WT-DXF2GEO.dll is used like this:

```
            try
            {
                DXF2GEO.Convert(DxfFilePath, GeoFilePath, tolerance);
DxfFilePath = Path to the DXF file e.g. C:\test.dxf
GeoFilePath = Path to the Geo file that is created e.g. C:\test.geo
tolerance = Max. distance of start/end point of two DXF Entities that belong together.
             }
            catch (FileNotFoundException e)
            {
                MessageBox.Show("Error: " + e.Message);
            }
The FileNotFoundException is always thrown if the DxfFilePath does not exist.
```

Create a tool list in WinTool: WT-MakeList

The WT-MakeList module allows the creation of a tool list in *WinTool*.
It is started with via an exchange file which contains a list of the *WinTool* tool assemblies and data about the tool list itself. A window appears in which the user can check and change the data.

The documentation and the module itself are in the folder "WT-MakeList".

History

WT-CAM-InterfaceApp

1.6.0.265

- Improved WorkAngle calculation
- Tool assembly description configuration

1.5.14174

- Fixed: Incomplete creation of the DXF files
- All data fields of the materials table are added to the xml file
- Updated calculation of datafield "ToolAssembly.CenterC"

1.4.13537

- Compatible with *WinTool* 2011 - 2014
- Separated program files and user data
- Updated calculation of datafield "ToolAssembly.TNumber"
- Updated usage of adjustment values

1.3.1.13095

- Added datafield "StockState" to XML

1.3.12830

- Compatible with WinTool 2013, 2012 and 2011
- Added datafields to XML
- Added configuration parameter set the decimal separator in the output XML
- Added latest versions of WT-ToolExport.exe and WTMakeList.exe
- Updated WT-DXF2GEO module

1.2.10863

- Added start parameter /I to WT-CAM- Interface-App.exe which exports a tool list by name
- Added new tool type "00"
- Improved error messages if interface is called with non-existent numbers or ToolTypes.xml has errors
- Added latest versions of WT-ToolExport.exe and WTMakeList.exe

1.1.1.10291

- ToolTypes.xml is searched in the interface startup path
- Corrected detection of the interface start parameters

1.1.9315

- Added/changed datafields in XML
- Integrated configuration window
- Included version nr. in XML
- Updated manual, created English version

1.0.8665

- First version

WT-ToolExport

2.4.2.128

- Displaying number of available Process-Duplicates in the list
- Displaying all duplicates per tool assembly
- Displaying default stock state of the assembly
- Resizable window
- Compatibility with *WinTool* 2018

2.3.13541

- Saving selection state of "preferred only" filter
- Improved readability with high DPI settings
- Compatible with *WinTool* 2014

2.2.11763

- Resizable search windows

2.1.10843

- Start-up time with large databases is quicker
- Selected work material and coolant type are displayed
- Parameters in .cfg file can be set via command line
- If a single start mode is set (only T, L or M), the corresponding search window is displayed directly
- "Current dataset"/"Total datasets" is now displayed above the tool/list/machine table
- Returning error codes
- Improvements of the user interface