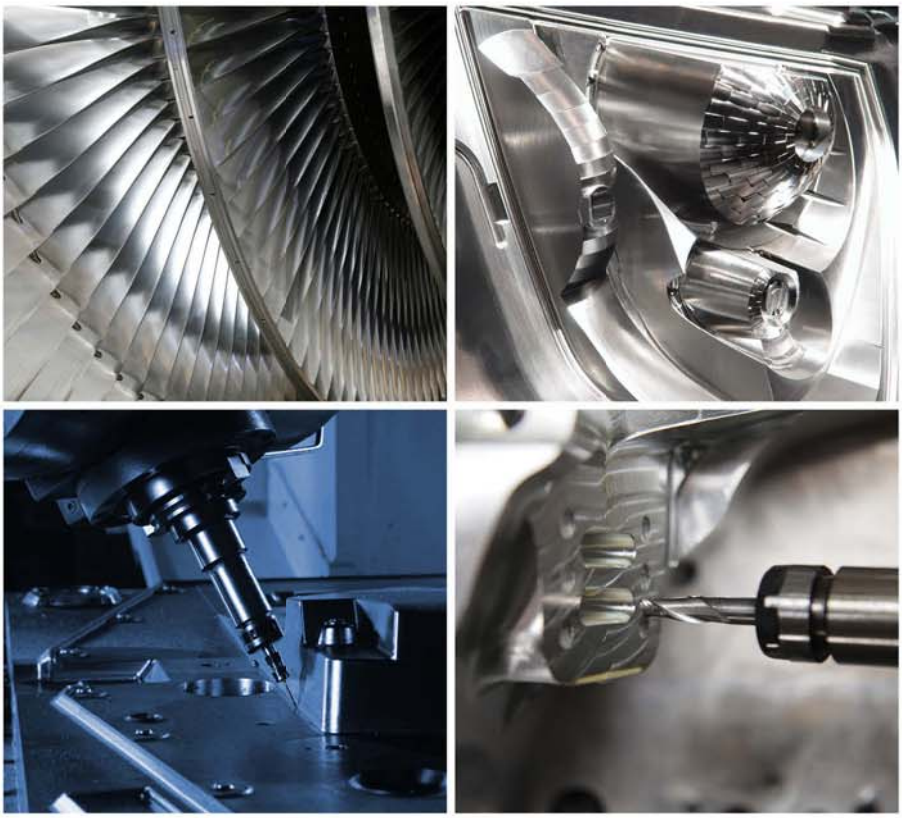
Delcam Advanced Manufacturing Solutions

Powering your productivity

PowerMILL



Training Course

Important Notice

This document is supplied as part of a Delcam Training Course. It is not intended to be distance-learning material: rather as an aid for Tutors when presenting material to course delegates and as a subsequent aid memoir to those delegates.

Delcam does not accept responsibility for any personal belongings / valuables whilst on the premises. Delegates are advised to keep their belongings on their person at all times.

Delcam plc. has no control over the use of the software described in this document and cannot accept any responsibility for any loss or damage howsoever caused as a result of using the software. Users are advised that all results from the software are checked by a competent person in accordance with good quality control procedures.

2016 Customisation Contents

Chapters	Page Number
1. Templates.	1.1 - 1.20
2. Basic Macros, User Menu.	2.1 - 2.10
3. Custom Toolbars	3.1 – 3.2
4. Tool Database.	4.1 - 4.8
5. Interactive Macros.	5.1 - 5.16

1. Templates

Strategy Templates

The default values/settings in a **Toolpath Strategy** dialog can be modified to those most likely to be required for a specific user's application.

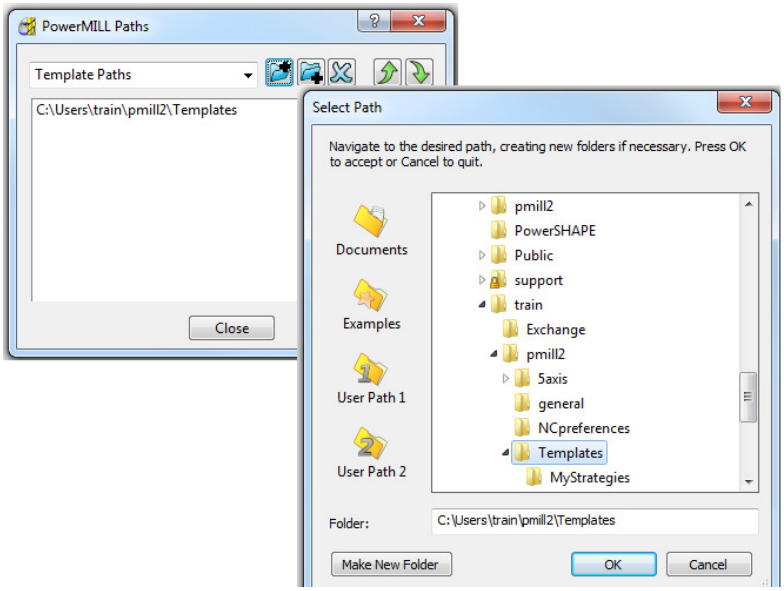
Each modified strategy will be saved as a **Template** accessible from a new (user defined) tab at the top of the **Strategy Selector** dialog.

Setting up a Strategy Template area

Once the set-up has been initiated, the user can easily update and save **Templates** to this chosen, area as and when required.

1. Select **File** > **Delete All** and **Tools** > **Reset Forms**.
2. From the ***Desktop*** open **Windows Explorer** and create the **Folder**:

   ….\pmill2\Templates\MyStrategies
3. From the main menus, select **Tools** > **Customise Paths**.
4. Select **Template Paths** from the pull-down menu.
5. Select the **Add path to top of the list** icon to raise the **Select Path** dialog.



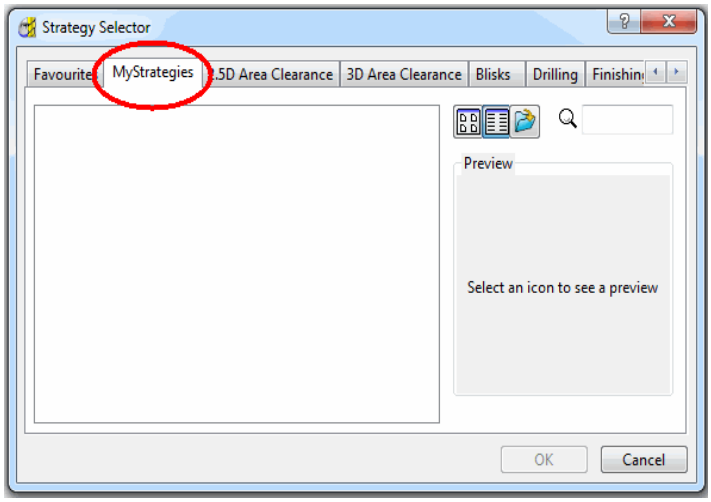
6. Browse to the folder location shown below for the required **Template Path**.

   In this case:- **….\pmill2\Templates**

7 Close the **PowerMILL Paths** dialog.

8 Select **Toolpath Strategies** .

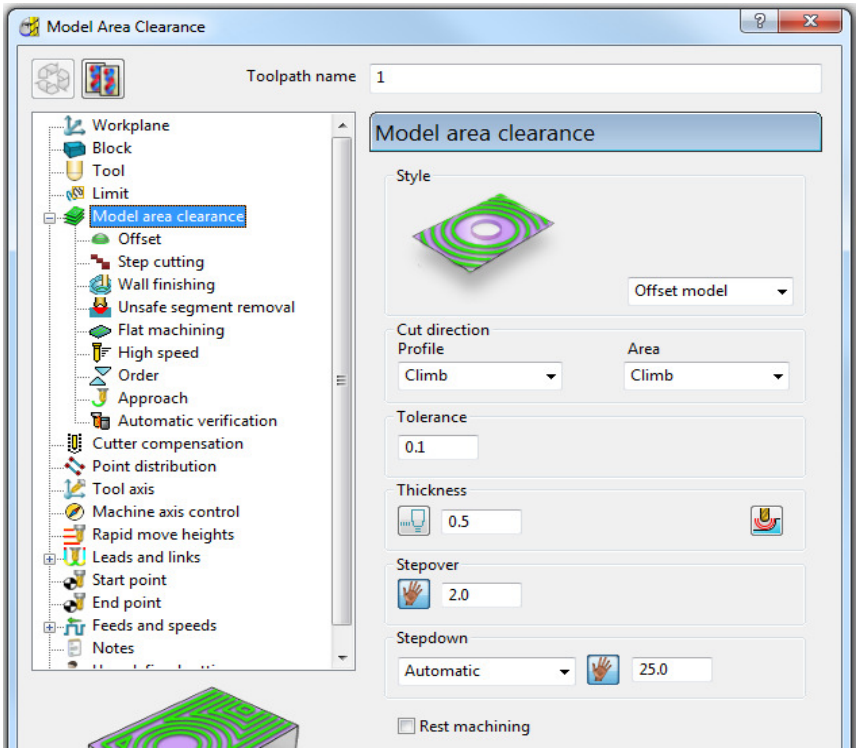
9 An additional tab called **MyStrategies** has been added to the dialog.



Creating Strategy Templates

1 Select **File** > **Delete All** and **Tools** > **Reset dialogs**.

2 Select the **Model Area Clearance** strategy from the **Toolpath Strategies** selector dialog.



3 Enter:

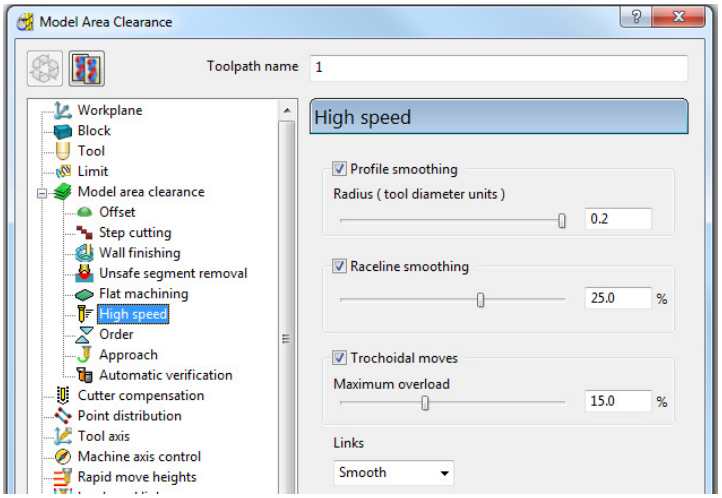
Thickness:- **0.5**

Stepover:- **2**

Stepdown:- **25**

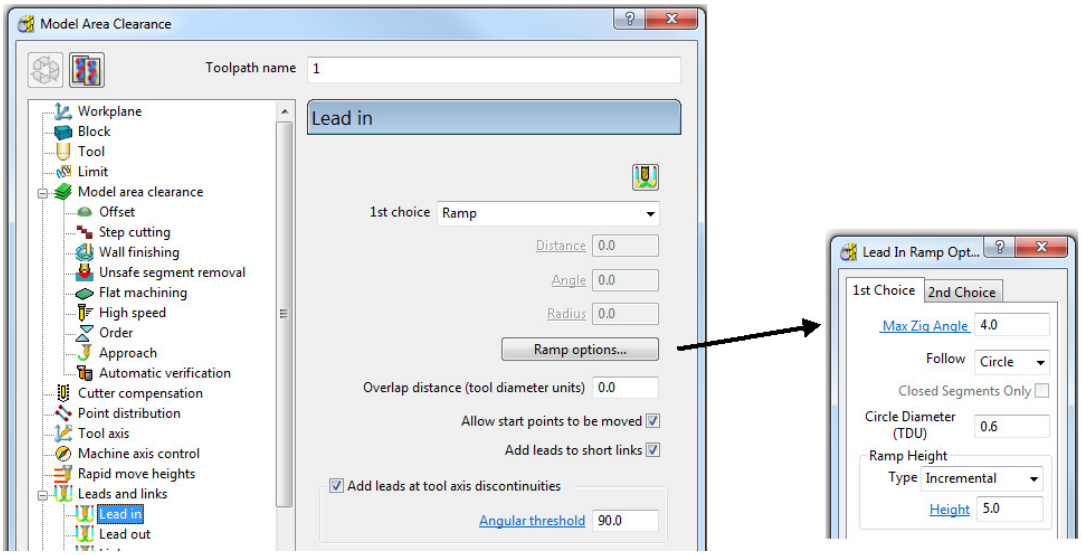
*Do not include a **Tool**, **Boundary**, or **Pattern** in the **Template Strategy**, otherwise **PowerMILL** will attempt to create an incomplete one every time the resultant **Template** is selected.*

1 Select the **High Speed** page from the local **explorer**.



4 Input the High Speed settings and values exactly as shown above.

5 Select the Lead in page from the local explorer.



6 Select the **Ramp options** tab and then in the **Lead In Ramp Options** dialog select the **1st Choice** tab.

7 In the same dialog Enter:

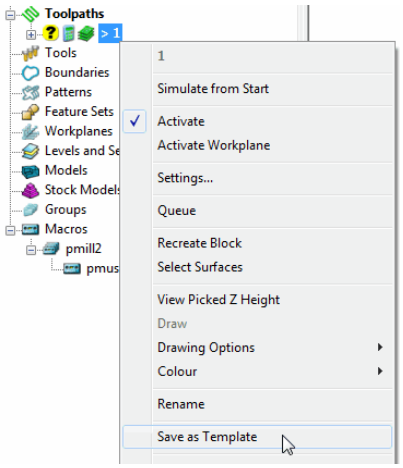
Follow:- **Circle**

Max Zig Angle:- **4**

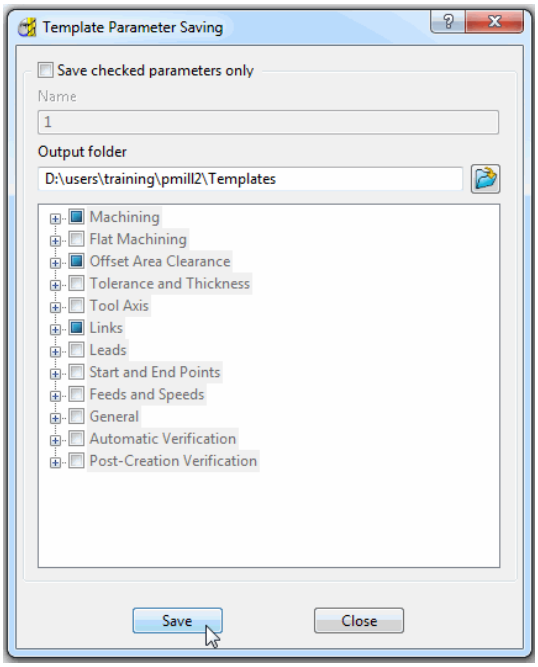
Circle Diameter:- **0.6**

8 Click **OK** (do not select **Calculate**) on the **Model Area Clearance** dialog.

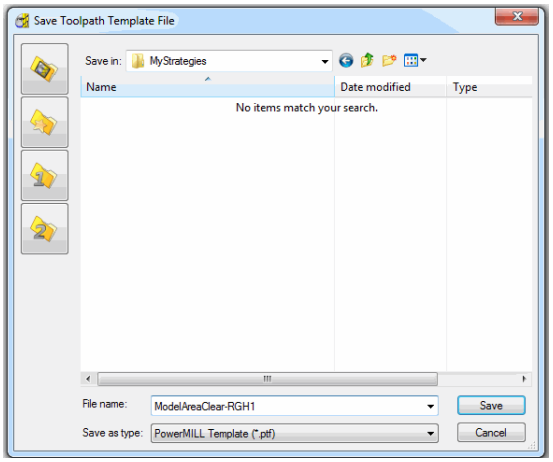
9 In the **PowerMILL explorer**, right-click on the un-calculated toolpath and from the local menu, select **Save as Template**.



10 Left-click on **Save** in the **Template Parameter Saving** dialog.

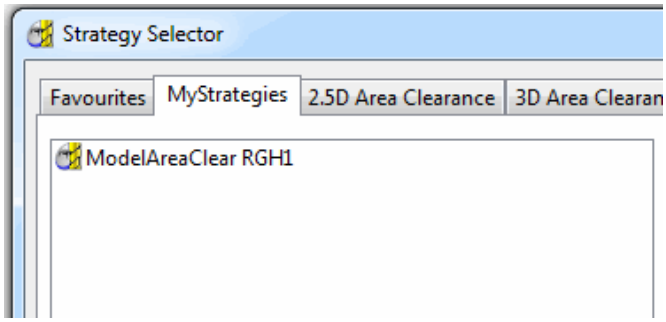


11 Double-click **MyStrategies**, type **ModelAreaClearRGH1** as the file name and click **Save**.



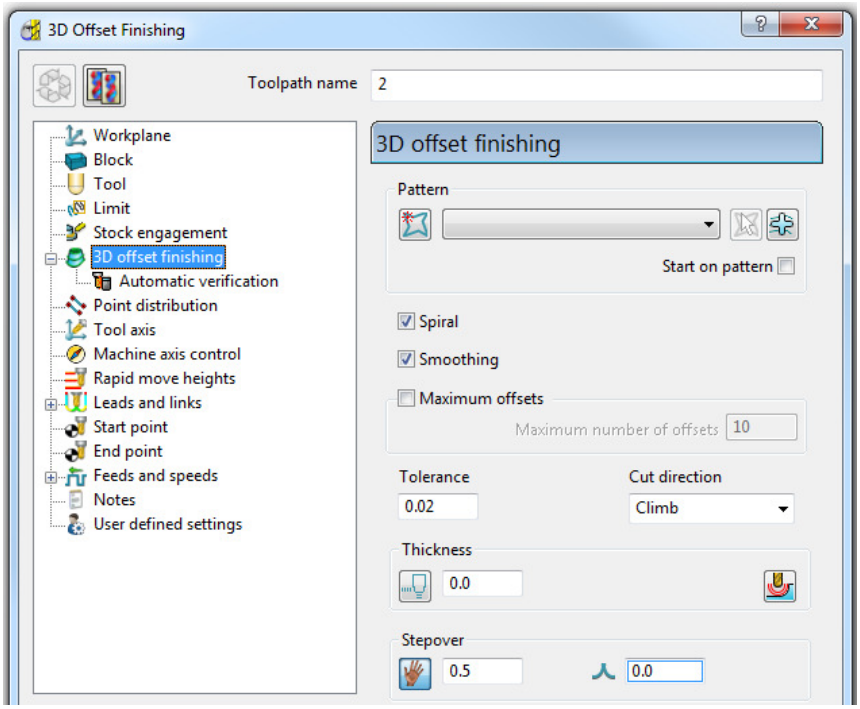
*Once the **Template** has been saved the dialog will close automatically.*

12 Select **Toolpath Strategies** .



In **Toolpath Strategies** > **MyStrategies** the new ***Template***, **ModelAreaClearRGH1** will now be available.

13 From the **Finishing** tab, select the **3D Offset Finishing** strategy.



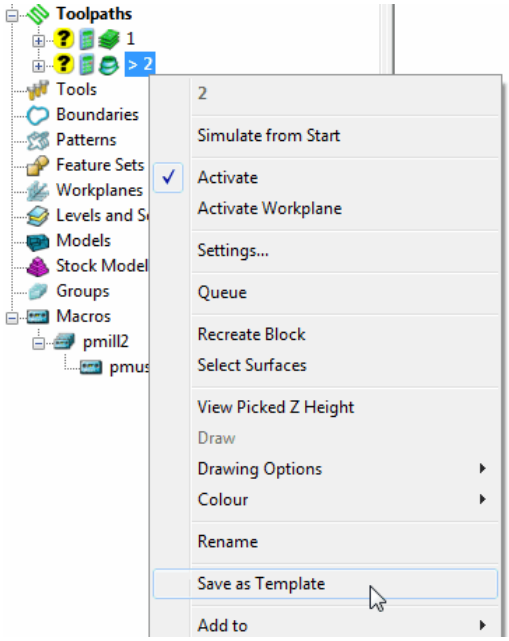
14 In the **3D Offset Finishing** dialog tick both **Spiral** and **Smooth** and Enter:

Tolerance: ***0.02***

Stepover: ***0.5***

15 Click **OK** (do not use **Calculate**) on the **3D Offset Finishing** dialog.

16 In the **PowerMILL *explorer***, right-click on the un-calculated **Toolpath** and select **Save as Template**.

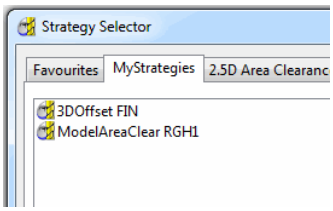


17 As before, right-click on **Save** in the **Template Parameter Saving** dialog.

18 Double-click **MyStrategies**, type **3DOffsetFIN** as the file name and click **Save**.

Once the Template has been saved the dialog will close automatically.

19 Select **Toolpath Strategies** .



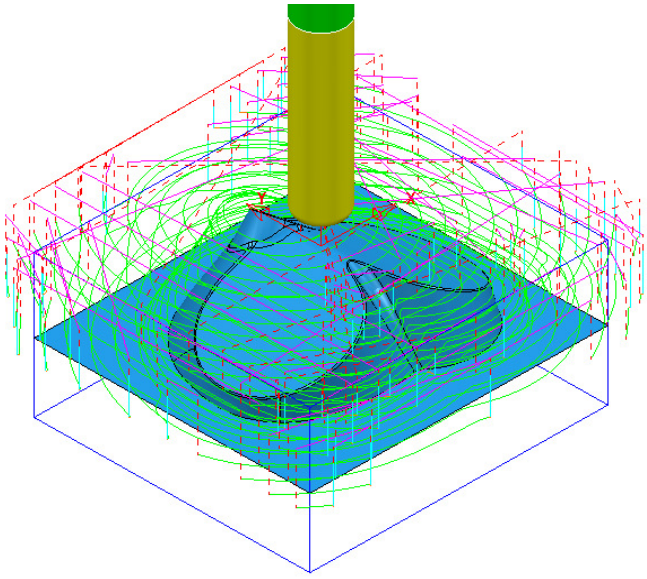
20 There are now two **Templates** in **MyStrategies**:- **ModelAreaClearRGH1** and **3DOffsetFIN**

Project Templates

In many applications the basic contents of a **Project** could be re-used as the basis for future jobs. These jobs would contain similar tooling, strategies, and associated settings. Any items that will differ from one **Project** to another must be deleted prior to saving it as a **Template**.

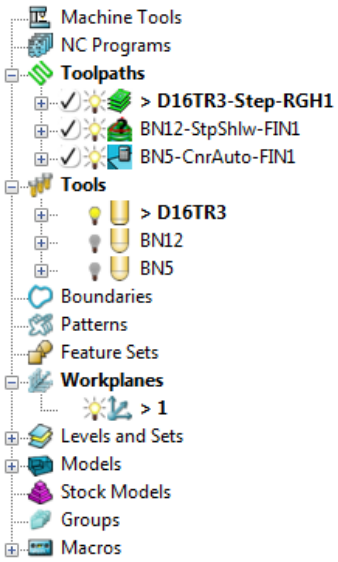
1 Select **File** - **Open Project**:-

….\PowerMILL_Data\Projects\ProjectTemplate_Start



2 Select **File** - **Save Project As**:-

….\COURSEWORK\PowerMILL_Projects\ProjectTemplate-1



The **Project** contains **Toolpaths**, **Tools**, a **Workplane**, and a component **Model**.

Both the existing **Workplane** and **Model** will be deleted and all **Toolpaths** ***Invalidated*** before a **Project Template** is saved.

3 Delete the **Model** from the current **Project**.

4 Try to **Delete** the **Workplane** but note that this is not currently possible as it is ***Locked*** to each **toolpath** and the **NC Program**.

5 In the **PowerMILL *explorer***, right mouse click on **Toolpaths** and from the local menu select **Invalidate All**.

6 Open the first **toolpath** and de-activate the **Workplane** before selecting **OK** to close the dialog (without processing the toolpath).

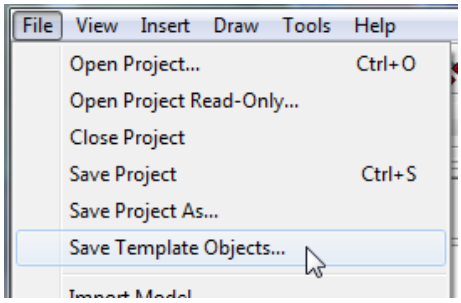
7 Repeat stage 6 on the other two **Toolpaths**.

8 Now delete the **Workplane** from the **PowerMILL *explorer***.

9 In the **PowerMILL *explorer***, right mouse click on **Levels and Sets** and from the local menu select **Delete Empty Levels and Sets**.

The **Project** now contains 3 un-processed **Toolpaths** and 3 **Tools**.

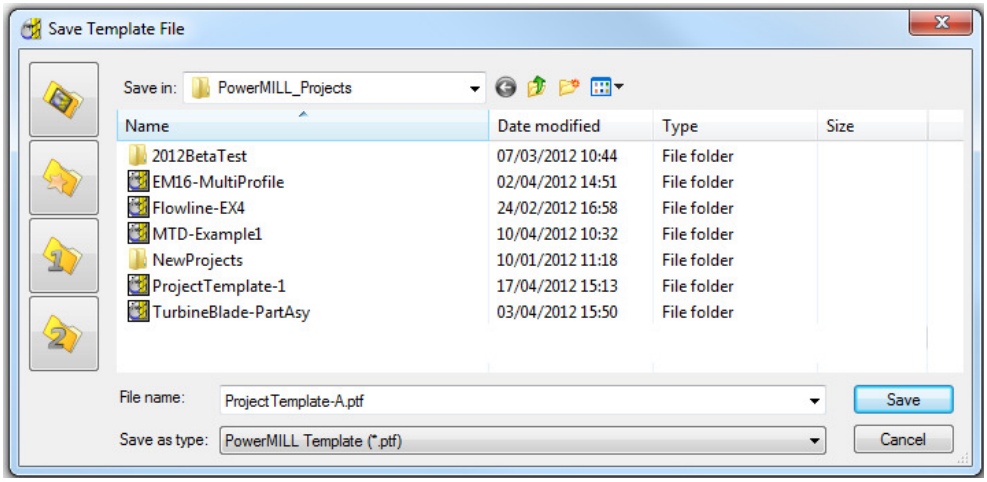
10 From the **main** pull down menus select **File** – **Save Template Objects**.



11 In the **Save Template** dialog:-

a ***Save in***: (Browse to) **...\COURSEWORK\PowerMILL_Projects**

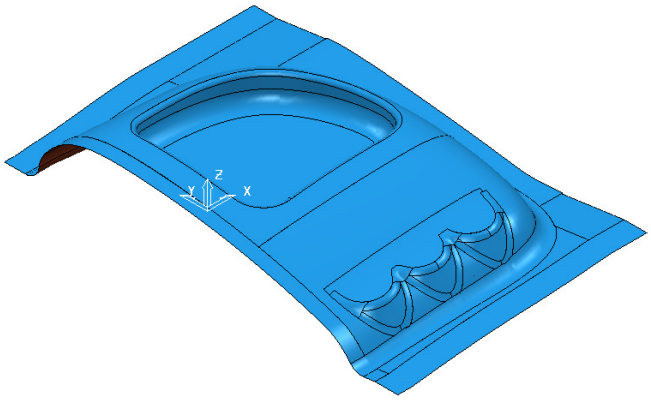
b Input ***File name***: **ProjectTemplate-A** before selecting **Save**.



12 From the **main** pull down menus select **File** - **Delete All** and **Tools** - **Reset Dialogs**.

13 **Import** the **model**:-

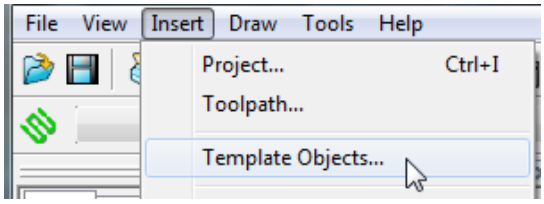
...\PowerMILL_Data\Models\facia.dgk



14 Select **File** - **Save Project As**:-

...\COURSEWORK\PowerMILL_Projects\Facia-UsingTemplate-1

15 From the **main** pull down menus select **Insert** > **Template Objects**



16 In the browser, select the **Template**:-

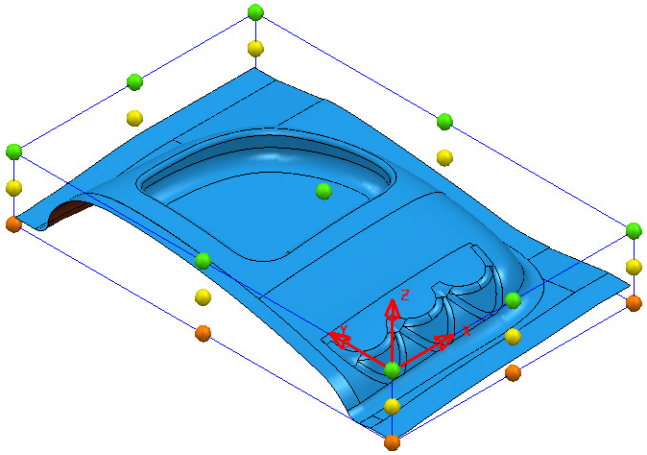
...\COURSEWORK\PowerMILL_Projects\ProjectTemplate-A.ptf

17 **Calculate** a **Block** defined by **Box** to the **Model** dimensions.

18 Select an **ISO1** view.

19 In the **PowerMILL explorer** right mouse click on **Workplanes** and select **Create and Orientate Workplane** > ...**Positioned using Block**.

20 Snap the sphere to the nearest top corner of the **Block**.



21 **Activate** the new **Workplane**.

The **Block** will move relative to the **Workplane's** co-ordinate system.

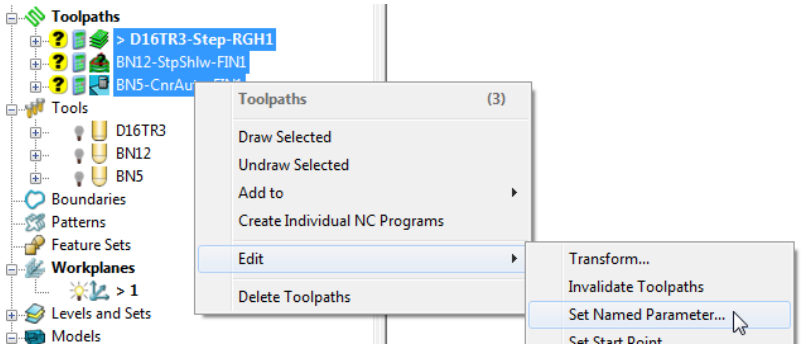
22 **Calculate** a **Block** defined by **Box** to the **Model** (and **Workplane**) dimensions.

Both the **Block** and **Workplane** need to be included in the un-processed toolpath strategies. This can be achieved collectively.

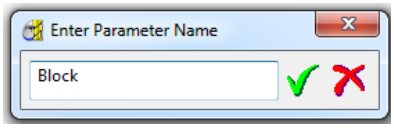
23 **Calculate** the **Rapid Move Heights** using the default settings.

24 In the **PowerMILL *explorer*** use the **left mouse** and **Shift** keys to select all 3 **toolpaths**.

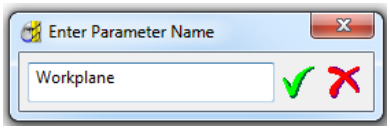
25 Right mouse click over the selected **toolpaths** and from the local menu select **Edit** > **Set Named Parameter**.



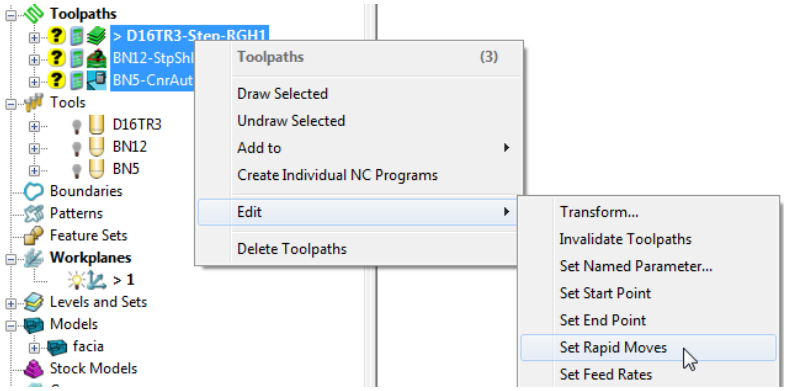
26 In the **Enter Parameter Name** dialog input **Block**.



27 Repeat stage **23** but this time input **Workplane** in the **Enter Parameter Name** dialog.

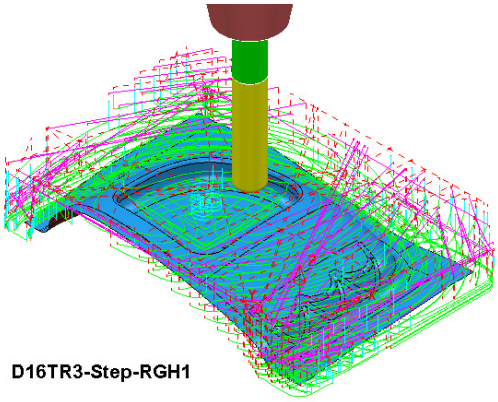


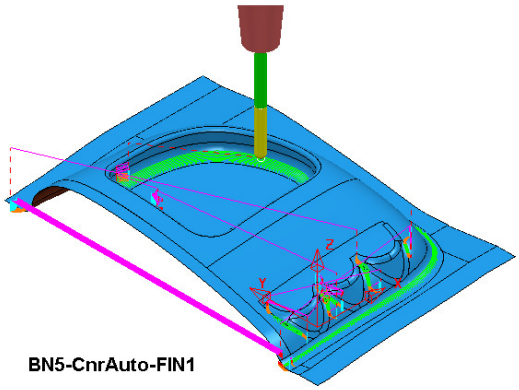
28 Right mouse click over the selected **toolpaths** and from the local menu select **Edit** > **Set Rapid Moves**.



29 In the **PowerMILL *explorer*** right mouse click on **Toolpaths** and from the local menu select **Batch Process**.

All the strategies will be processed to include the added **Block**, **Workplane**, and **Rapid Move** settings.





30 Select **File** - **Save Project**:-

(....\COURSEWORK\PowerMILL_Projects\Facia-UsingTemplate)

31 From the **main** pull down menus select **File** - **Delete All** and **Tools** - **Reset Dialogs**.

Exercise

1 **Import** the **model**:-

...\PowerMILL_Data\Models\phone.dgk

2 Select **File** - **Save Project As**:-

...\COURSEWORK\PowerMILL_Projects\phone-UsingTemplate-2

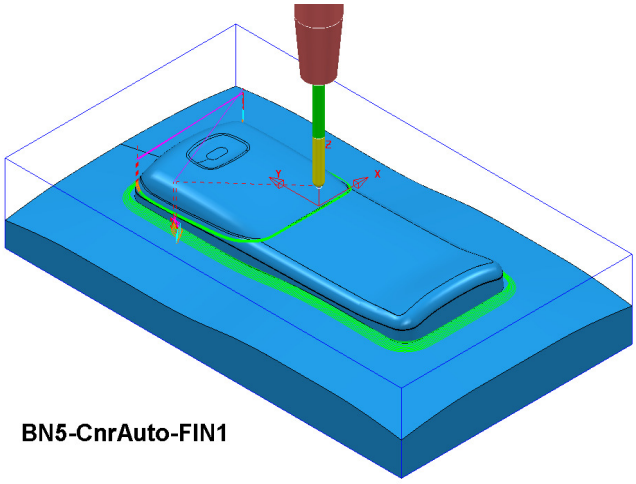
3 Create and **Activate** a **Workplane** at the ***Top Centre*** of the **model**.

4 **Calculate** a **Block**, and reset the **Rapid Move Heights**.

5 **Insert** the **Template:-**

...\COURSEWORK\PowerMILL_Projects\ProjectTemplate-A.ptf

6 Collectively set up and **Batch** process the **Toolpaths**.



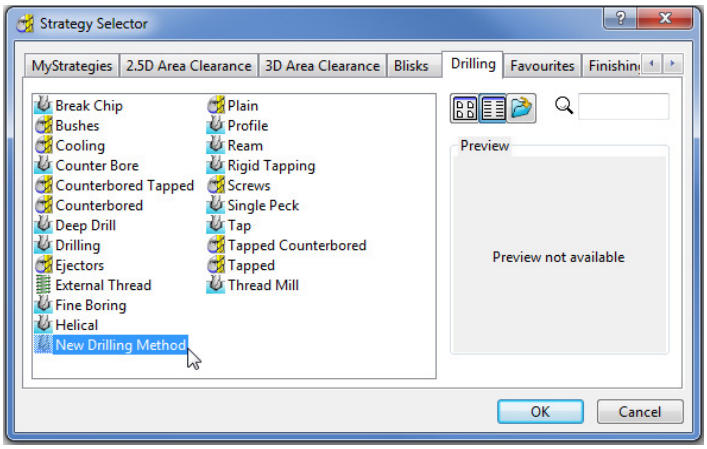
Drilling Methods

A **Drilling Method** is a specialised **Template** that defines a range of drilling strategies that will be assigned to specific **Hole** types. Initially you determine a selection criteria (which holes do you want to drill) followed by a set of drilling strategies (how do you want to drill them). This sequence can be repeated for a different selection of holes.

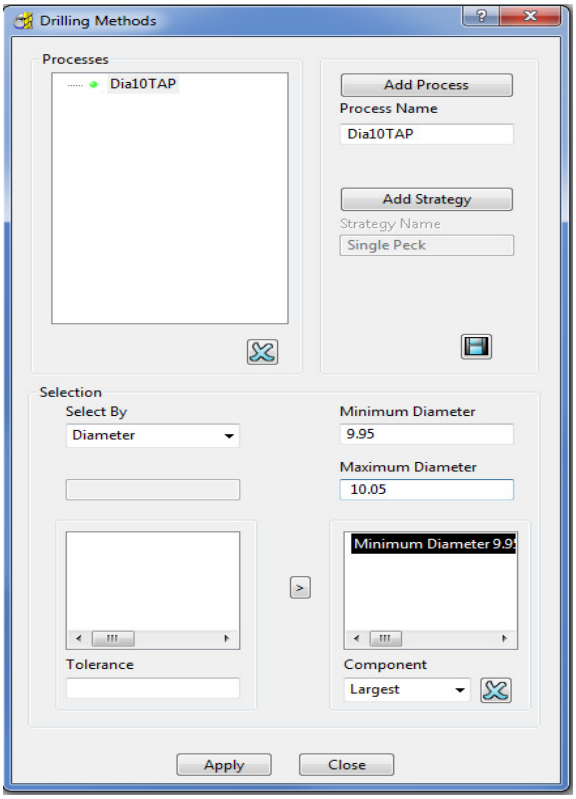
1 Select **File** - **Delete All** and **Tools** - **Reset Dialogs**.

2 Select the **Drilling** tab from the **Strategy Selector** dialog.

3 Select the **New Drilling Method** option from the menu then **OK**.

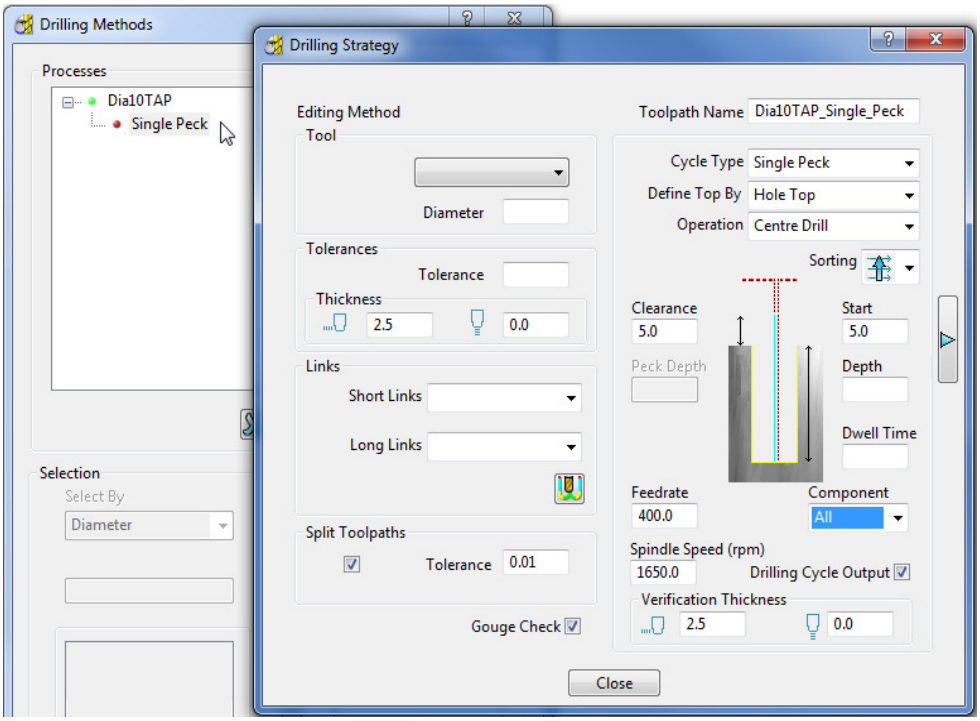


4 Select **Add Process** and input the ***Process Name*** as **Dia10Tap**.



5 Click on **Dia10TAP** in the ***Processes*** window and then ***Select By*** – **Diameter** with the above values (min **9.95** max **10.05**).

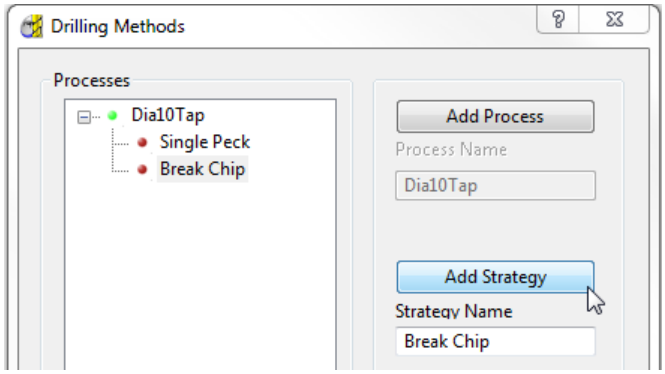
6 Click on **Add Strategy** from the **Drilling Methods** dialog (This opens the **Drilling Strategy** dialog defaulted to **Single Peck**).



7 Enter the values into the dialog exactly as shown above with the **Sorting** option set to **Along the Y direction, one way in X**. 

This **Single Peck** operation will identify all holes within the min/max range specified, dia **10mm** combined with a **Radial Thickness** of **2.5mm** in both the ***Thickness*** and ***Verification Thickness*** fields. This will result in all dia **10mm** holes being interpreted as dia **5mm** holes for the tool selection.

8 **Close** the **Drilling** dialog.

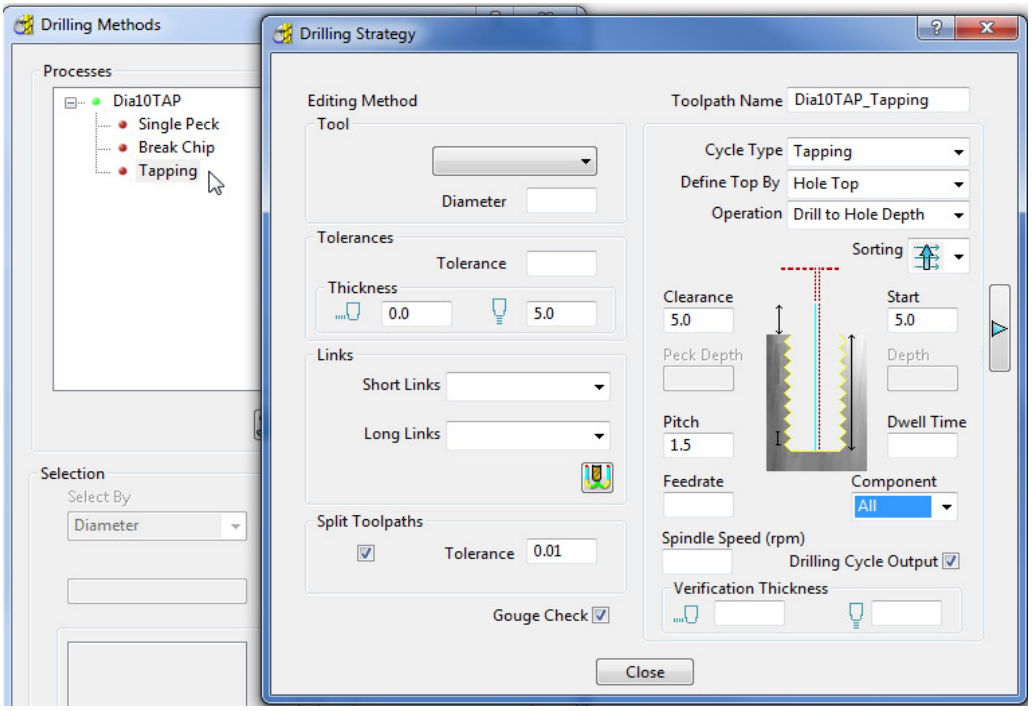


9 Select **Add Strategy** and in the **Drilling Strategy** dialog input:- ***Cycle Type*** - **Break Chip**, ***Operation*** – **Full Diameter**, and **Radial Thickness 0.75**.

10 **Close** the dialog.

Before the **Tapping** operation the holes are to be drilled with a **Dia 8.5 Drill**. This **Break Chip** operation will identify all holes within the **min/max** range specified, dia **10mm** combined with a **Radial Thickness** of **0.75mm** in both the ***Thickness*** and ***Verification Thickness*** fields. This will result in all dia **10mm** holes being interpreted as dia **8.5mm** holes for the tool selection.

11 Select **Add Strategy** and in the **Drilling Strategy** dialog input:- ***Cycle Type*** – **Tapping**, ***Operation*** - **Drill to Hole Depth**, **Axial Thickness 5**, and **Pitch 1.5**.

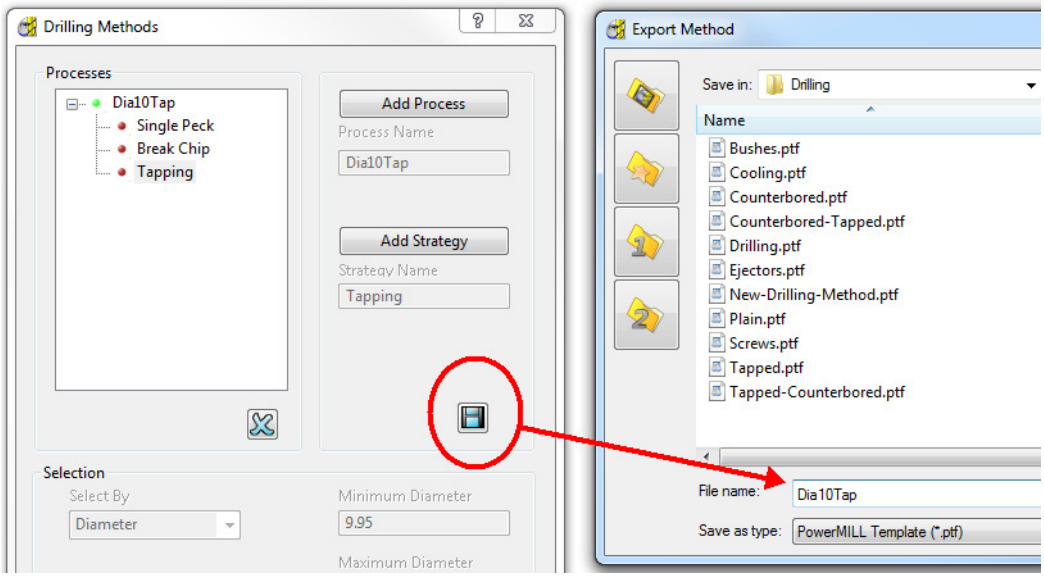


12 **Close** the **Drilling Strategy** dialog.

13 Select the **Save** option from the **Drilling Methods** dialog. 

14 For **PowerMILL** to recognise the method in the **Tool Strategies** - **Drilling** dialog it must be saved with a suitable name (**Dia10Tap**) in the directory:-

C:\Program Files\Delcam\PowerMILLxxxx\file\templates\Drilling

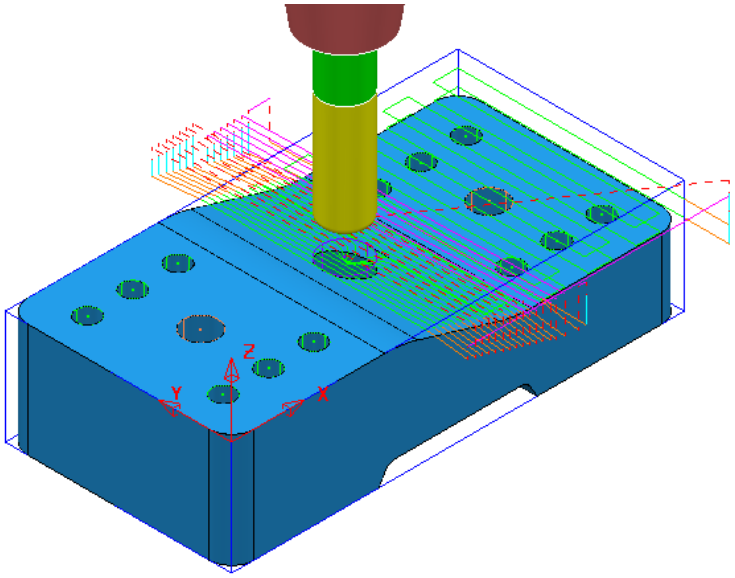


15 **Close** the **Drilling Methods** dialog.

16 Select **File** - **Delete All** and **Tools** - **Reset dialogs**.

17 Select **File** - **Open Project**:-

...\PowerMILL_Data\Projects\DrillingMethod_Start

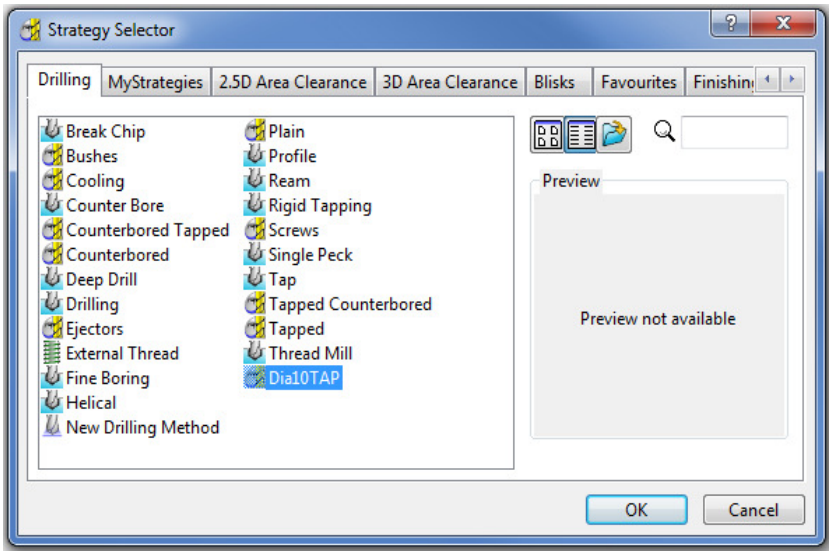


18 Select **File** - **Save Project As**:-

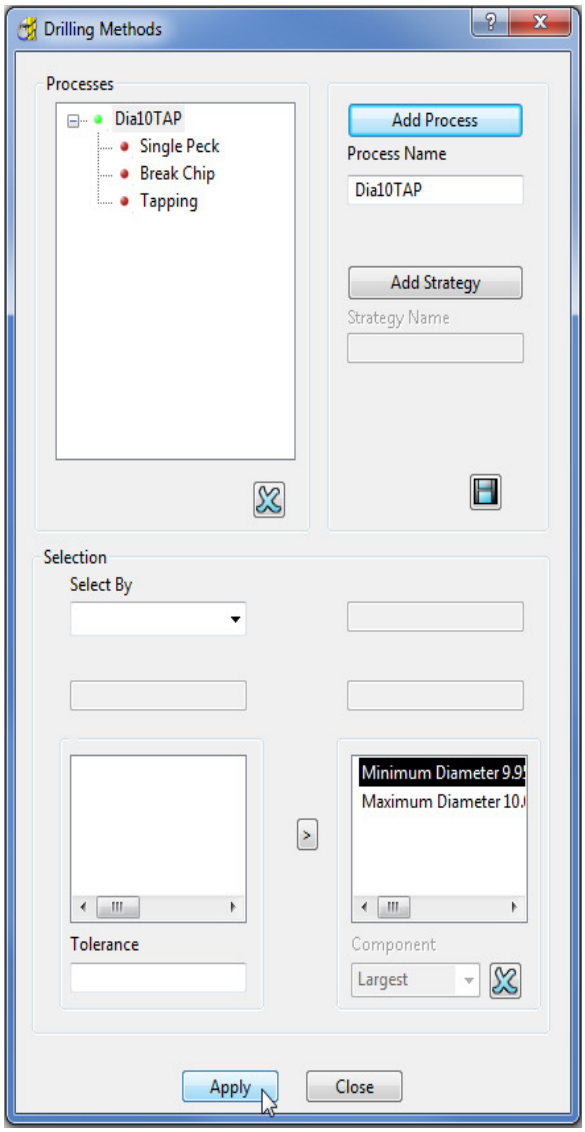
...\COURSEWORK\PowerMILL_Projects\DrillingMethods-EX1

The **Project** contains **Dia 8.5**, **Dia 5 Drills**, and an **M10-Tap** will be identified and used by the **Drilling Method** saved earlier.

19 Select the **Drilling** tab from the **Toolpath Strategies** dialog.

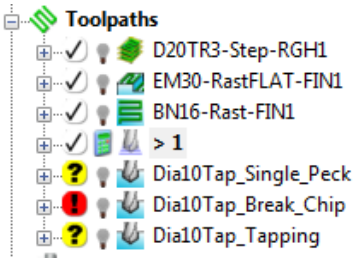


20 Select the newly created, **Drilling method** (**Dia10Tap**) as displayed in the above dialog and click on **OK**.



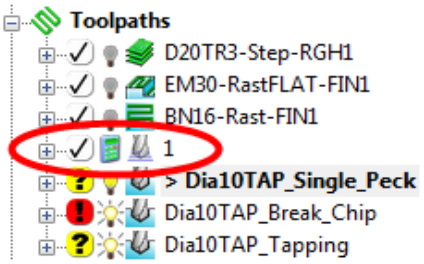
21 Select **Apply** at the bottom of the ***Drilling Methods*** dialog.

A set of Drilling/Tapping strategies are created on the Dia 10 hole **Features**.



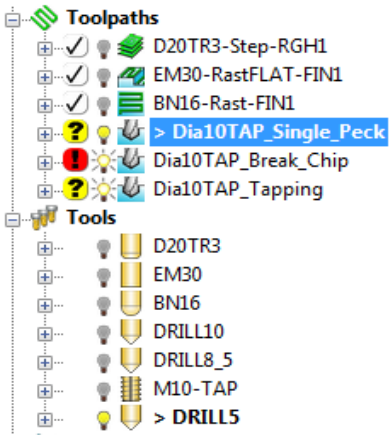
22 **Simulate** each of the 3 Drilling/Tapping **Toolpaths**.

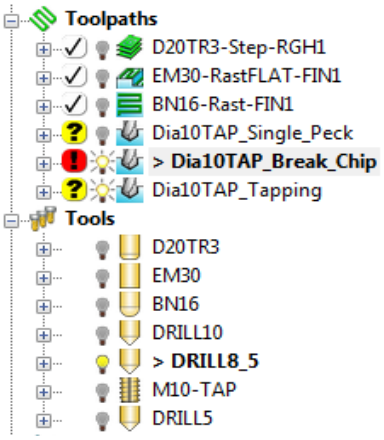
Note that the **Drilling Method** (**Dia10Tap**) has made the tool selection based on the stored information. The **Single Peck** has used a **Dia 5 Drill**, the **Break Chip** has used a **Dia 8.5 Drill** and the **Tapping** has used an **M10-TAP**.



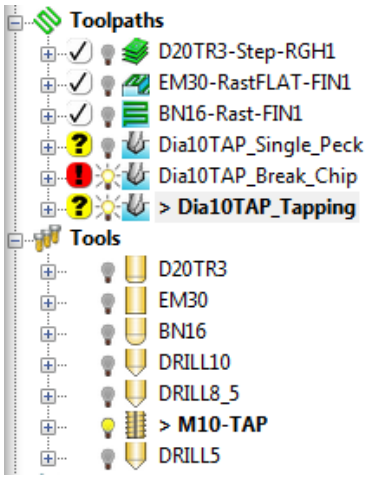
23 **Select** and **Delete** the uncalculated **Drilling Toolpath 1** created as a consequence of the **Drilling Method** being applied.

24 In the **PowerMILL *explorer***, **Activate** the ***toolpath*** **D10TAP_Single_Peck** and the tool **DRILL5** used will also become **Active** (As shown below left).





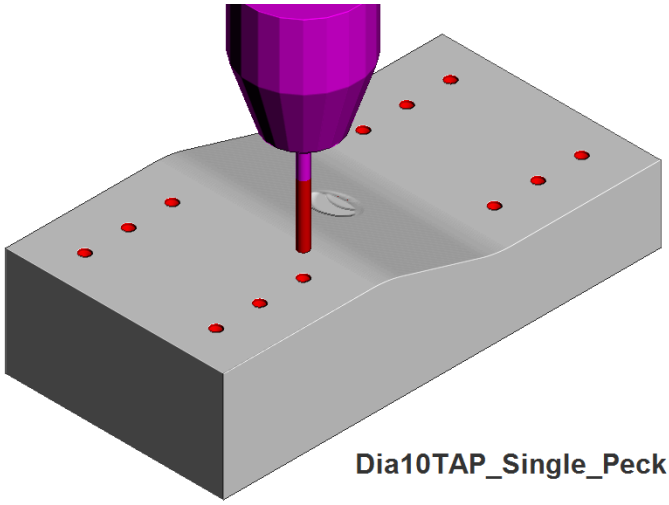
25 In the **PowerMILL *explorer***, **Activate** the ***toolpath*** **D10TAP_Break_Chip** and the tool **DRILL8_5** used will also become **Active** (As shown above right).



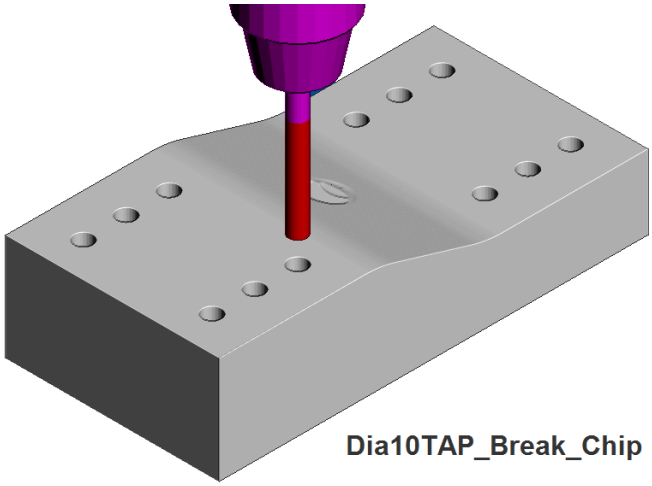
Only the holes with diameters within the selection range (**DIA 9.95-10.05**) have been drilled and tapped. The actual **Drill** size selected is further controlled by inserting **radial thickness** values into the individual ***Drilling Process*** strategies.

26 Select **File** - **Save Project**.

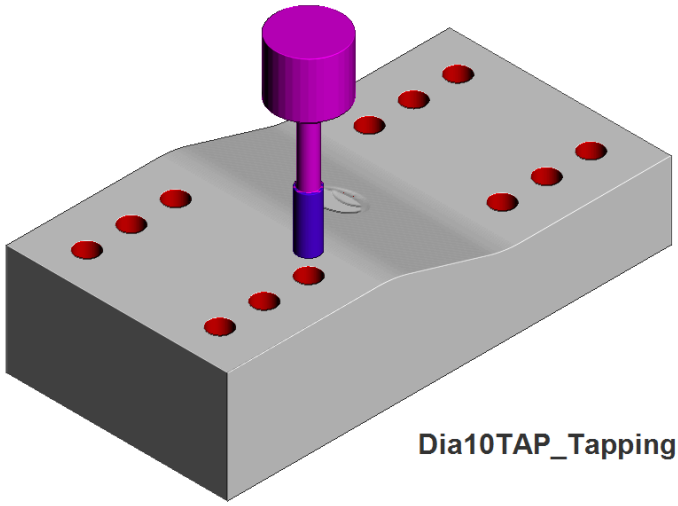
27 Run a full **ViewMILL simulation** on all 6 **toolpaths** in the **Project**.



Dia10TAP_Single_Peck



Dia10TAP_Break_Chip



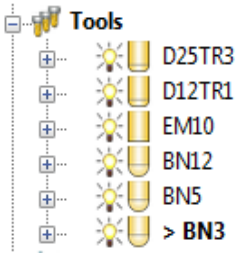
Dia10TAP_Tapping

Tool Library Templates

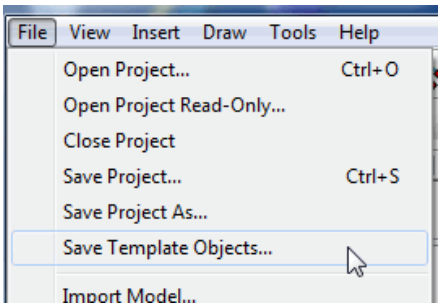
As well as being applied to customise machining strategies, **Templates** can also be created for other entities such as the definition of a tool library. A major advantage of applying **Templates** for tool libraries is the ease at which they can be updated and re-saved. It is also quicker to **Insert** a **Template** than, to run a **Macro**.

1 Select **File** > **Delete All** and **Tools** > **Reset Dialogs**.

2 In the **PowerMILL *explorer*** create the following tools including suitable **Shank/Holder** definitions and **Feeds/Speeds**.

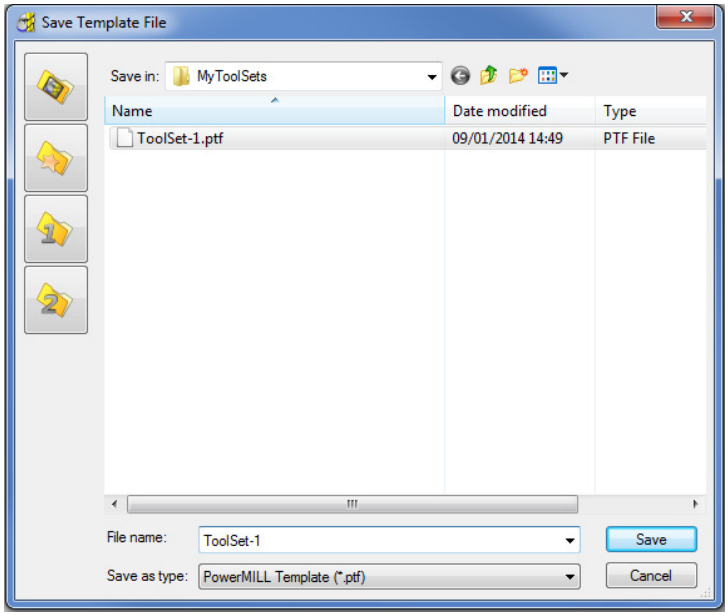


3 From the **File** pull down menu, select the option **Save Template Objects**.



4 In the **Save Template File** dialog browse to:

C:\pmill2\MyToolSets



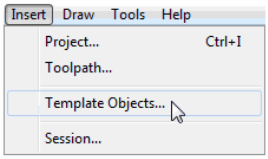
*Do not save **Tool Library Templates** to the same area as **Toolpath Strategy Templates** as this will cause operational problems.*

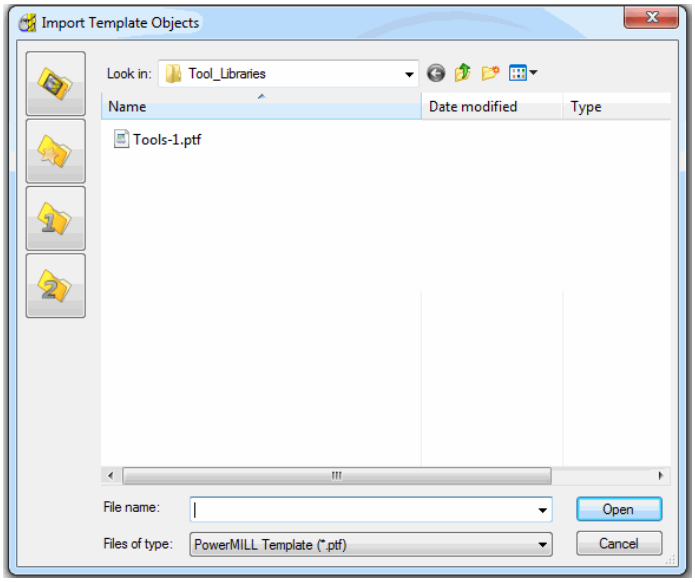
5 Input the file name **Tools-1** and click **Save**.

A major advantage of using **Templates** to input tooling into **PowerMILL** is that individual tools can quickly be modified and the Toolset updated by clicking on the **Save** tab.

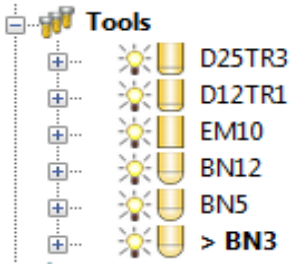
6 **Delete** all of the **Tools** currently stored in the **PowerMILL *explorer***.

7 From the **Main** pull-down menus select **Insert** > **Template Objects** and open the **ToolSet-1.ptf** template file in the **Import Template Objects** dialog.





8 All of the **tools** stored in the ***template*** will immediately appear in the **PowerMILL *explorer***.



A series of **Tool Library Templates** can be created for such variants as different **Machine Tools** and **Material Types**.

This method of **Tool Definition** can be applied to each **Project** and any unused **tools** can, if required be deleted from the **PowerMILL *explorer***.

*The above **Tools** could also be stored as separate **Templates Objects** to be imported into the **PowerMILL Project** as individual items.*

2. Macros & User Menu

Introduction

There are various ways in which the user can customise **PowerMILL**. This chapter looks at the creation of:

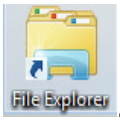
- **Macros**
- **User Menus**

Home Area

The usual method to enable the **User Menu** to be recognised by **PowerMILL** is to set up and use a **home** area in ***Windows***. This can be located to any convenient area on the hard disk. In the following example the **home** area is:-

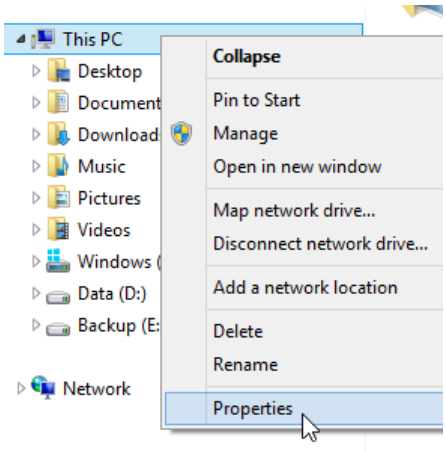
C:

To check if a **Home** area already exists on a computer;

1 Open **File Explorer** 

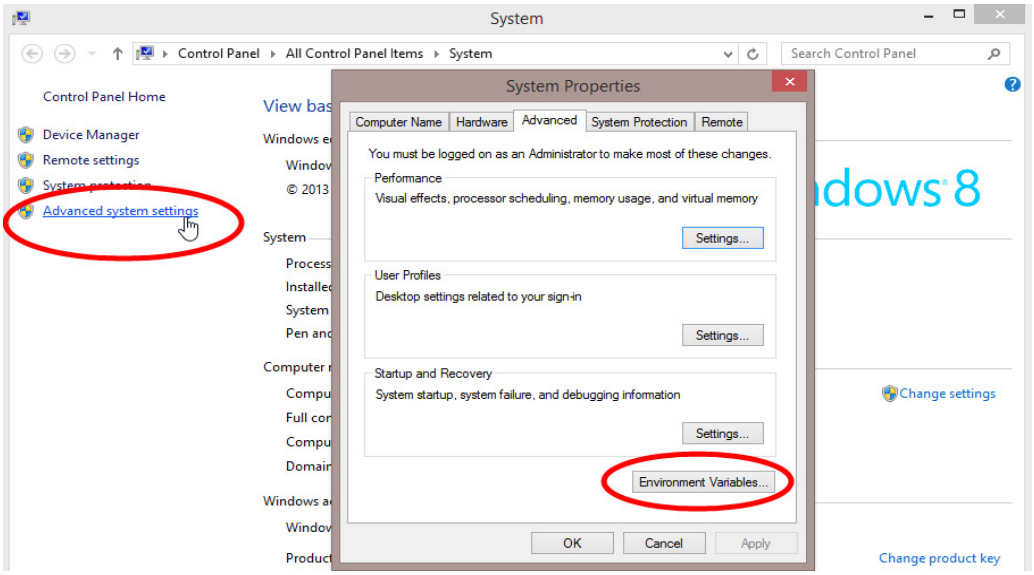
.

In the local explorer, right-click on **This PC** and select **Properties** to open the **System** dialog.

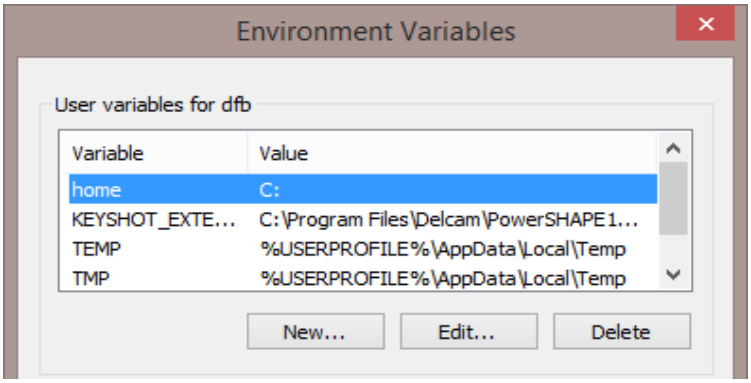


Alternatively, at the lower left corner of the screen, right mouse click on ***Start*** *and from the local menu select* ***System****.*

2 In the **System** dialog select the **Advanced system settings** option and then in the **System Properties** dialog, select **Environment Variables**.



3 If the **home** area is not already defined as shown below, select **New...**



4 In the ***User Variables*** dialog, enter **home** in the ***Variable*** name field and **C:** in the ***Value*** field.

5 Click **OK**.

You must have Admin rights to create a ***New User Variable****.*

6 If it does not already exist, create a folder named **pmill2** in the required **home** area (**C:\pmill2**).

If the ***pmuser.mac****,* ***user_menu*** *and any other* ***macro****s are being stored in this folder, then* ***PowerMILL*** *will have immediate access through the default* ***Macro*** *search paths.*

Macros

A **Macro** is a text file, which contains a sequence of commands to drive **PowerMILL** operations. These can be created by recording each command or by typing in the commands directly. **Macros** (which have a **.mac** extension) can then be run from the **PowerMILL *explorer*** or from a ***user defined menu***.

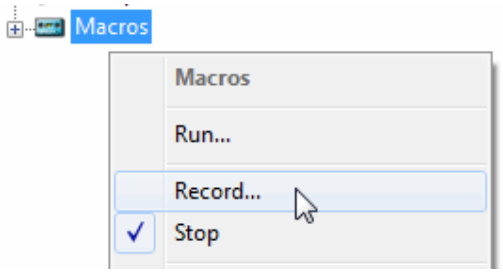
When **PowerMILL** starts up, it will attempt to identify and run the **initialisation macro** (if it exists!) called **pmuser.mac**. By adding customised **PowerMILL** commands to the initialisation macro, the user can set up their own default settings and parameters.

For immediate use a **pmuser.mac** can be placed in the **pmill2** folder, directly below the users **Home** area. The **pmill2** directory is also a convenient location for storing other user-defined **macros** ideally arranged in suitable sub-folders. This will be covered later.

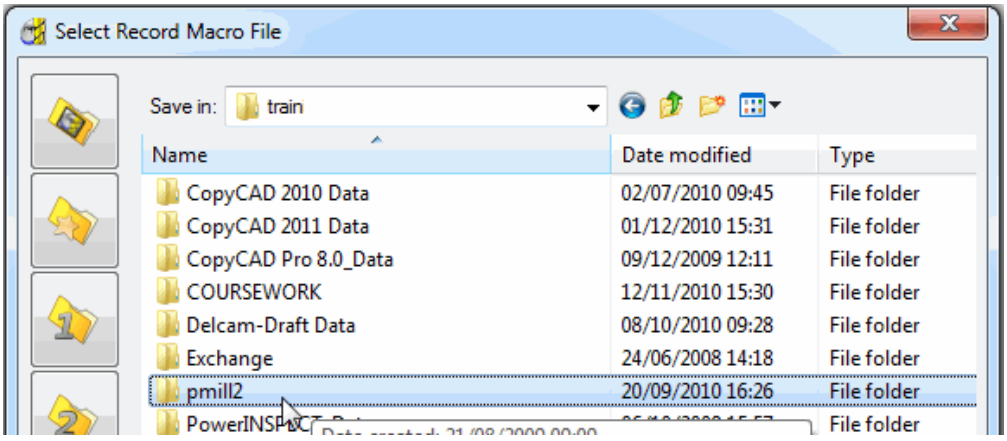
Creating a pmuser macro

Macros are created in **PowerMILL** by recording the operations as they occur. Only flags or parameters that are physically changed will be recorded into the macro; therefore to record a value that is already set, it must be re-entered or a flag re-switched. For example, if the finishing tolerance is currently set to 0.1mm, and it is required that the macro stores the same value, then it must be re-entered over the existing copy in the form during recording.

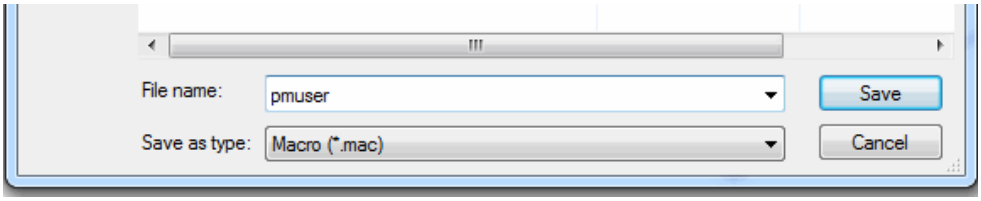
1. Select **File** > **Delete All**.
2. To start recording, right-click on **Macros** in the **PowerMILL *explorer*** and select **Record**.



3. Save **pmuser.mac** by first clicking on the existing folder displayed in Save in.



4. Then, in the resultant dialog, browse to **C:\pmill2** and at the bottom of the dialog enter the file name, **pmuser** before clicking **Save**.



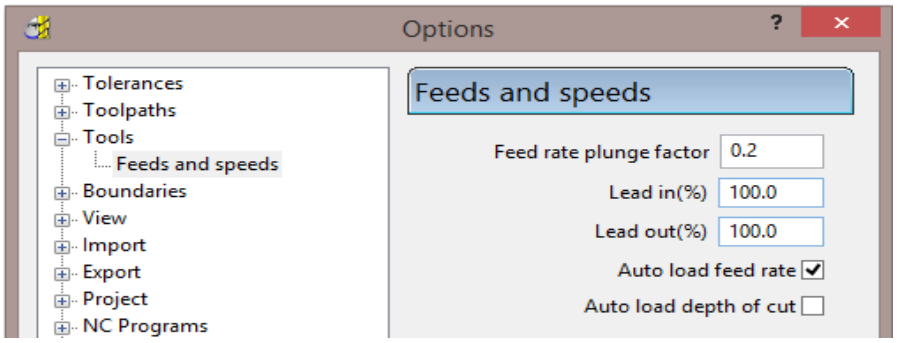
The Macro icon changes to red while recording is in progress

The **pmuser.mac** automatically runs whenever **PowerMILL** starts up, thereby providing the user with custom default settings. It can also be run at any time during a session to reinstate the initialisation settings without having to restart **PowerMILL**.

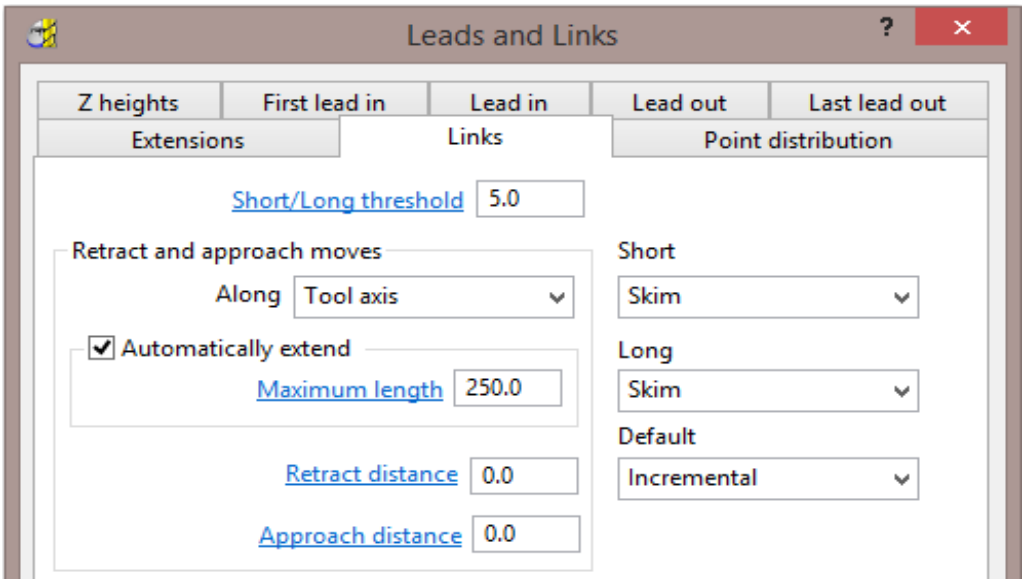
All required data or options must be set or selected if they are to be included in a macro (Even if it is necessary to overwrite or switch them off first).

5 From the **Main** pull-down menus select **Tools** > **Options**.

6 Select **Tools** > **Feeds and Speeds**, change the **Feed Rate Plunge Factor** to **0.2**, tick the **Auto Load Feed Rate** box, and **Accept** the dialog.

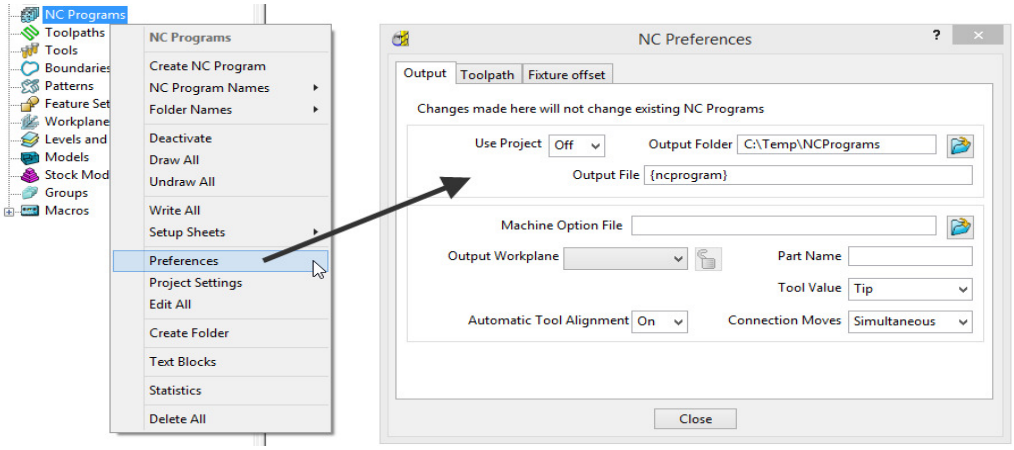


7 Open the **Leads and Links** dialog and input **Short/Long Threshold** as **3**, both ***Short*** and ***Long*** as **Skim**, and ***Default*** as **Incremental**.



8 **Accept** the **Leads and Links** dialog.

9 In the **PowerMILL *explorer***, right-click on **NC Programs**, and select **Preferences**.



10 On the **NC Preferences** dialog **Output** page, select the **Machine Option File** as **heid400.opt**.

11 Even if already set, select:

Use Project - **On**

Tool Value - **Tip**

Automatic Tool Alignment - **On**

Connection Moves - **Move, Rotate**

12 Select the **Toolpath** page and even if already set, select:-

Tool Change - **On New Tool**

Tool Numbering - **As Specified**

Tool Change Position - **After Connection**

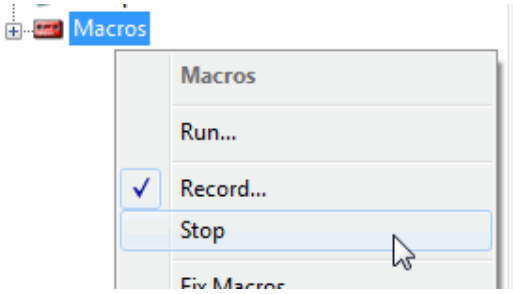
Cutter Compensation: **Length - Off**

Radius - **None**

13 Select **Apply**, followed by **Accept**.

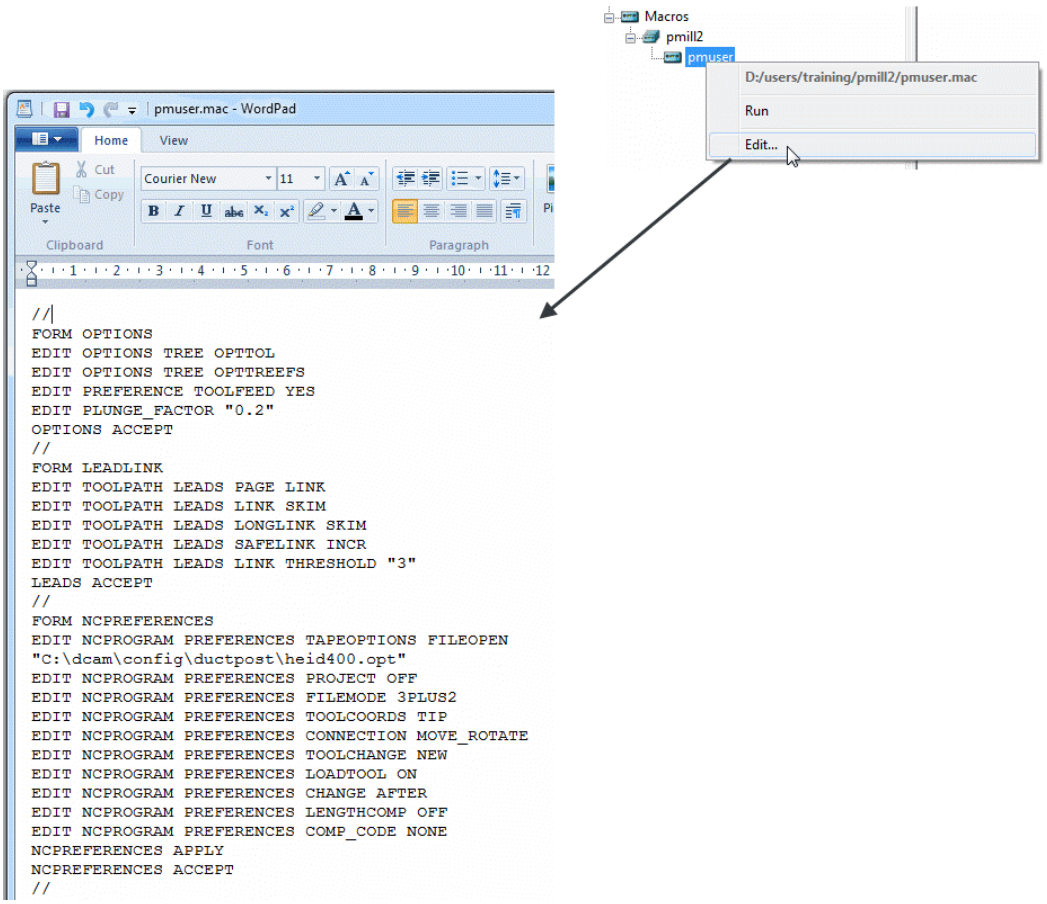
*It is important to remember to **Accept** and not to **Apply** forms that are opened while creating an **Initialisation Macro**, the only exception being the creation of **NC Program Preferences** (An initialisation file is for pre-setting forms but at this stage without executing the commands).*

14 Select **Stop** from the local **Macros** menu to stop the recording process.



15 Select the plus (+) sign next to **Macros** in the **PowerMILL *explorer*** to open the tree.

16 Right-click **pmuser**, then select **Edit**.



This is a standard **Wordpad** document that can be edited with new values and then saved.

To try this out you will need to exit and re-open **PowerMILL**.

17 Select **File** > **Exit** and restart **PowerMILL** to check that the settings from **pmuser.mac** have been activated.

Other Typical Macro Applications

Apart from tailoring **PowerMILL** by the creation of an initialisation macro other useful applications include macros for un-draw, draw and resetting **Leads and Links**, locating and inserting tooling **Templates**, setting **NC Preferences**, regularly used machining sequences or settings, and input of preview, commands not yet available on the GUI. Note the pmuser macro can be run through the browser at any time during a **PowerMILL** session to restore your settings.

Creating NC Preference macros

1 From the **PowerMILL *explorer***, record a **macro** called **Fanuc6m_prefs** and save it in:-

C:\pmill2\NCpreferences\...

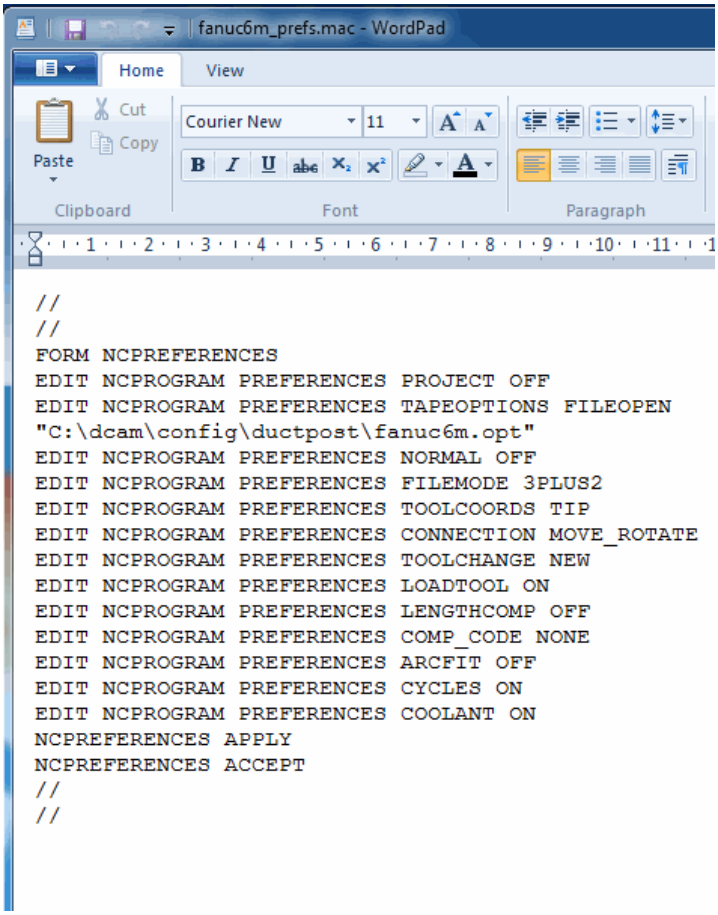
2 In the **NC Programs** > **Preferences** > **Output** tab select the **Machine Option File** as **fanuc6m.opt**.

3 Input the remaining settings exactly as set earlier for the **Heid400** in the pmuser macro.

4 Stop the **Macro** recording process.

Even if already set it is essential to re-specify settings if they are to be recorded into the macro.

A **macro** file (**Fanuc6m_prefs.mac**) is now created that sets up **NC Preferences** for a post-processor option file and the command lines are shown below.




```
//
//
FORM NCPREFERENCES
EDIT NCPROGRAM PREFERENCES PROJECT OFF
EDIT NCPROGRAM PREFERENCES TAPEOPTIONS FILEOPEN
"C:\dcam\config\ductpost\fanuc6m.opt"
EDIT NCPROGRAM PREFERENCES NORMAL OFF
EDIT NCPROGRAM PREFERENCES FILEMODE 3PLUS2
EDIT NCPROGRAM PREFERENCES TOOLCOORDS TIP
EDIT NCPROGRAM PREFERENCES CONNECTION MOVE_ROTATE
EDIT NCPROGRAM PREFERENCES TOOLCHANGE NEW
EDIT NCPROGRAM PREFERENCES LOADTOOL ON
EDIT NCPROGRAM PREFERENCES LENGTHCOMP OFF
EDIT NCPROGRAM PREFERENCES COMP_CODE NONE
EDIT NCPROGRAM PREFERENCES ARCFIT OFF
EDIT NCPROGRAM PREFERENCES CYCLES ON
EDIT NCPROGRAM PREFERENCES COOLANT ON
NCPREFERENCES APPLY
NCPREFERENCES ACCEPT
//
//
```

Exercise

5 Create two more independent **NC Preference** macros for the Hied400 and Mazak option (Use the filenames **h400_prefs.mac** and **mazak_prefs.mac** respectively for the macros), and save them to:

C:\pmill2\NCpreferences\...

User Menu

In **PowerMILL** it is possible to create a ***user menu*** enabling the user to further streamline and customize the command structure. To access the ***user menu***, right mouse click within an empty area of the **PowerMILL *explorer***.

1 Right-click in an empty area of the **PowerMILL *explorer*** to access the current **user_menu** (if it exists).

This menu is created from a text file called user_menu. It must be in the directory **pmill2** in the **Home** area of the computer for the individual user, and saved as a text file (not a word Document) and with no extension. If available it is easier to obtain a copy of an existing user_menu and modify it as required.

A **user menu** can be setup to provide easy access to existing **PowerMILL commands**, Macros and Templates.

2 Load the in-active **user_menu** file (which is in the **pmill2** folder) into **Wordpad**.

Below are instructions for the **user_menu** command line input (these exist in the sample file).

```
# ************** USER MENU ***************
# comment lines
# this file must be called user_menu and be in your HOME area

# This file must be called user_menu and must be located in a directory named
# pmill2 below the user's HOME area.
# T is for Title and cannot be selected.
# S is for separator or line on the menu.
# I is a command line containing a "menu instruction" - code no. - "action".
# M is start new menu/sub-menu.
# Z is end menu/sub-menu.
# comment lines.
# format is I "Label String" 1 "Command string"

U user_menu
T "[My Menu]"
S
```

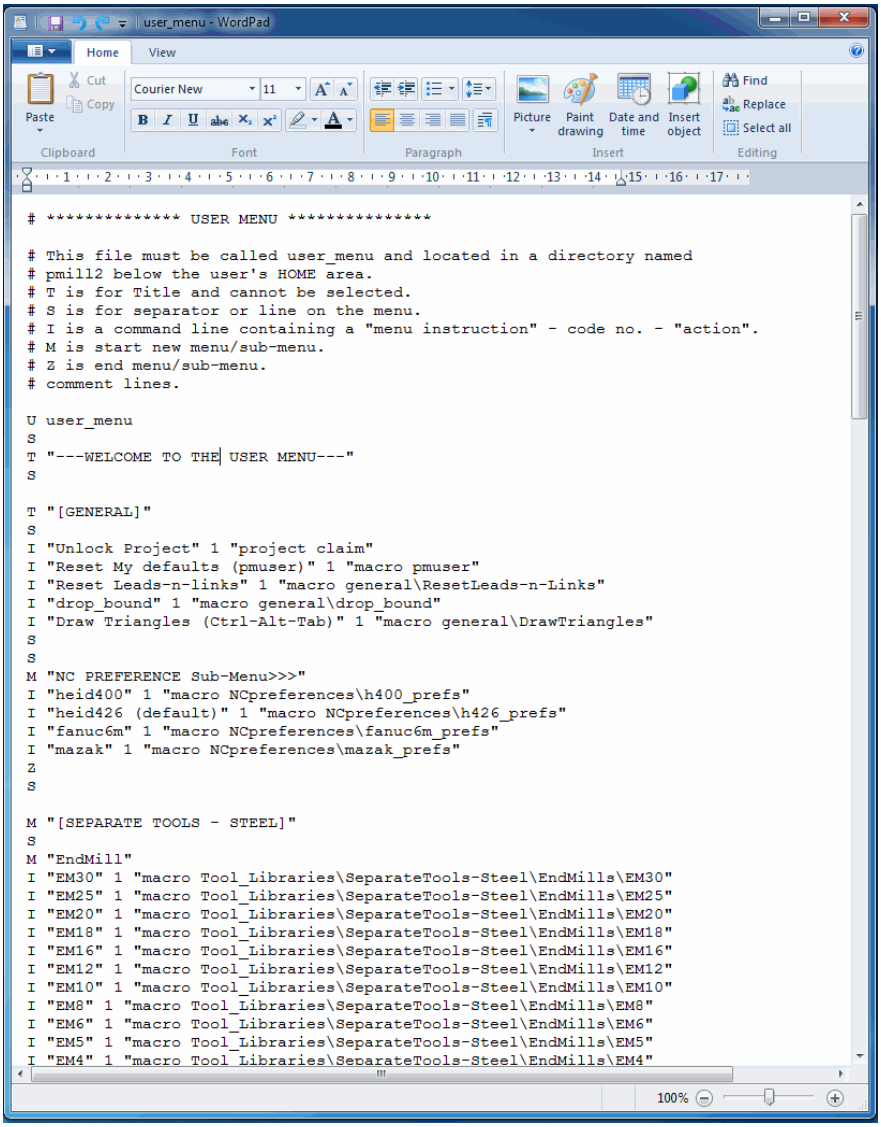
A command line will be added to the above user_menu to run the **h400_prefs** macro.

```
I "heid400" 1 "macro C:\pmill2\ncpreferences\h400_prefs"
```

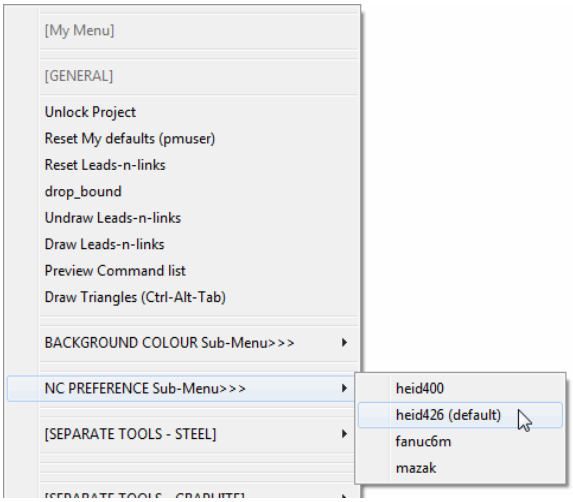
3 Add the above command line to run the **h400_prefs** macro directly from the **user_menu**.

This line locates and runs the file **h400_prefs.mac** created earlier in this chapter. It needs to include the correct drive and path where the macro is located. All lettering must be in the correct upper or lower case and include the inverted commas and hyphens.

A more complete version of a typical **user_menu** is as shown below:



This is the **user_menu** file as displayed in the **Wordpad**.



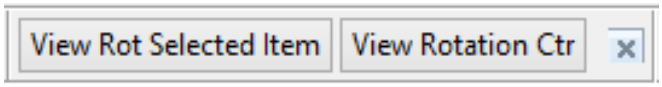
Note the **NC-Preferences** section uses a ***sub-menu***:

M denotes the start of the sub-menu.

Z denotes the end of the sub-menu.

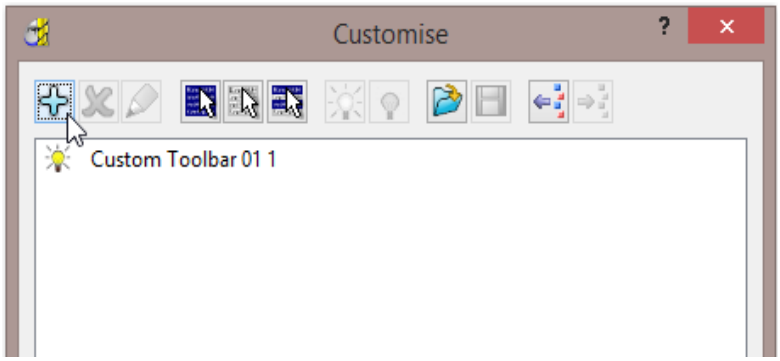
3. Custom Toolbars

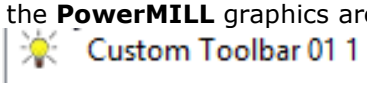
It is possible for the user to create **Custom toolbars** (example shown below). These can be set to run; **Menu**, or **toolbar** options, **Macros**, direct **PowerMILL** commands.

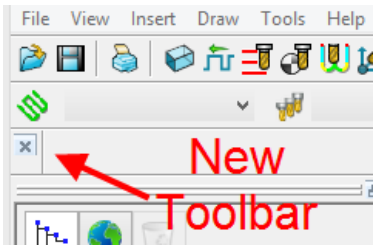


The above example provides quick access to switching the **View** - **Rotation Anchor** between being located at the centre of pre-selected items (**Selected items**) or at the screen centre (**View**).

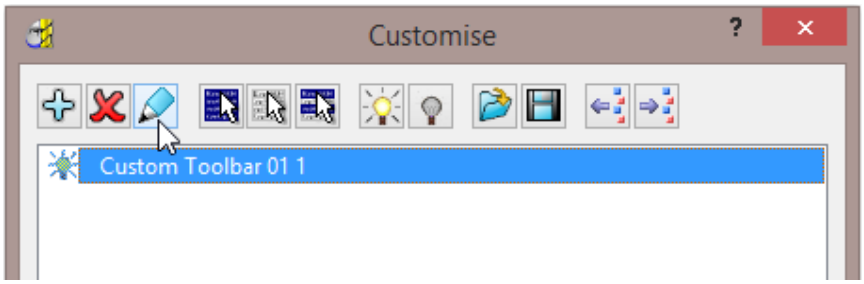
1 From the main pulldown menus select; **View** - **Toolbar** - **Custom** - **Customise**.



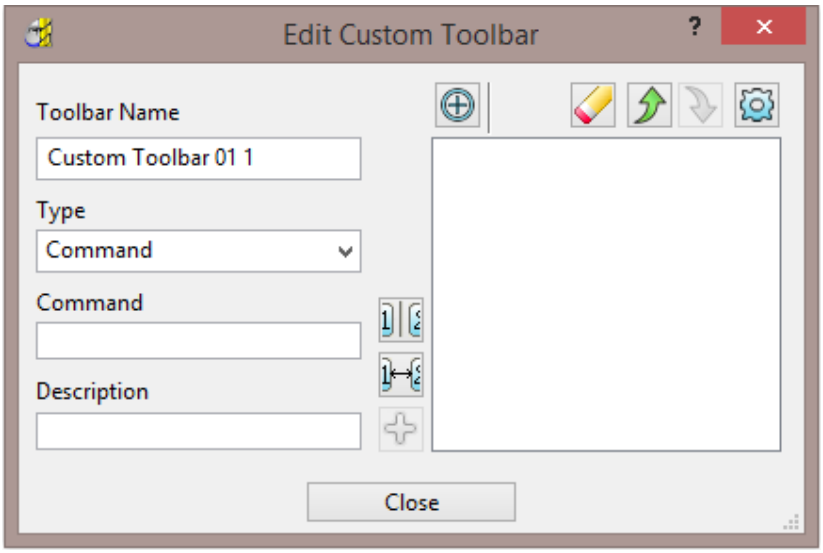
2 The above dialogue appears and if the  is clicked, a new (empty) **Custom Toolbar** (as shown above) will appear both in the **PowerMILL** graphics area, and as listed in the above dialogue window. 



Once the new **Custom Toolbar** is created a series of command actions can be added to it as required.



3 Select the new **Custom Toolbar** (Left Mouse Click) in the ***Customise*** dialogue and then select the '**Edits the selected toolbar**' icon. 

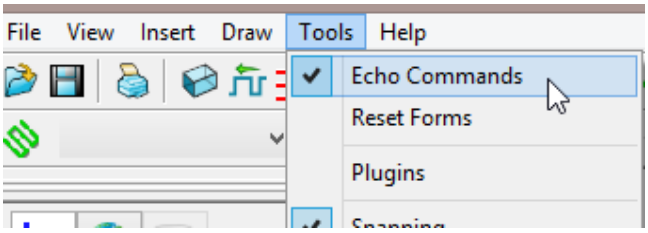


4 In the above dialogue, select ***Type*** – **Command**.

The equivalent text based, PowerMILL command for the **View** - **Rotatation Anchor** to be on a **selected item** is now inserted into the ***Command*** field.

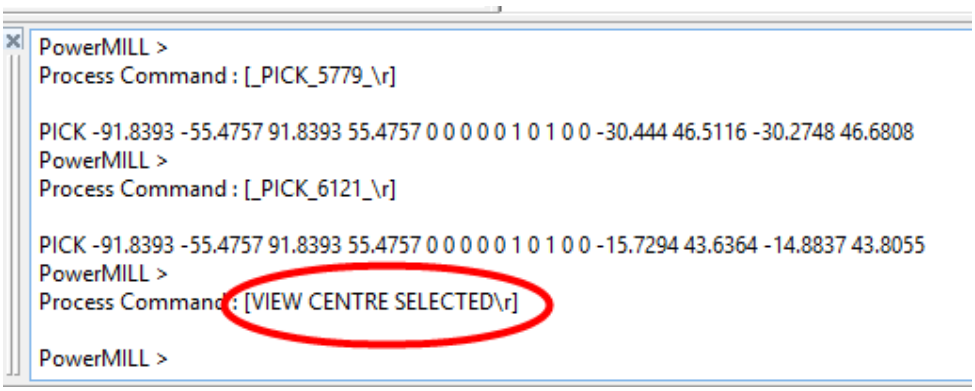
To obtain the correct syntax the relevent command can be ***echoed*** into the PowerMILL **command** window to be ***Copied*** and ***Pasted***.

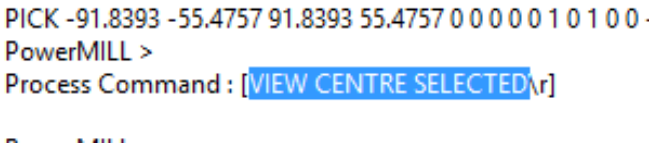
5 From the main pulldown menus select; **Tools** – **Echo Commands**.



The **Command** input window will appear below the **PowerMILL** graphics area.

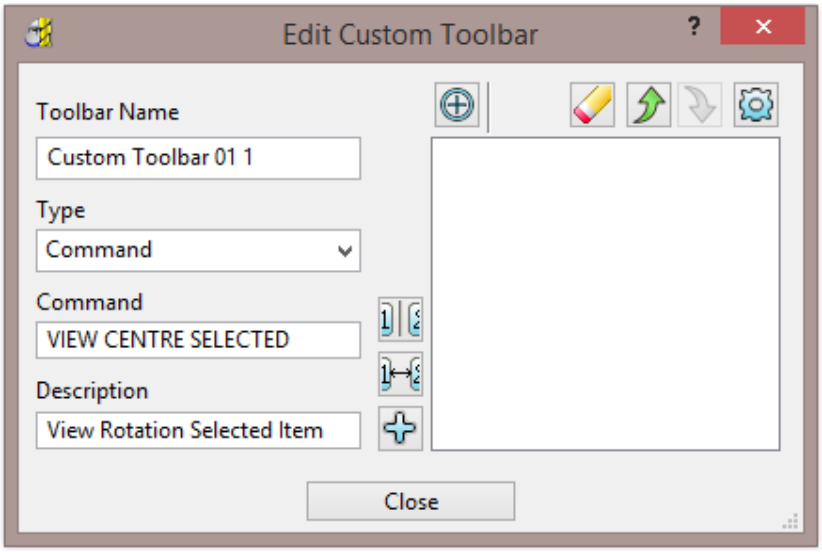
6 From the main pulldown menus select; **View – Rotation Anchor – Selected Items** and the equivalent text based version will be echoed into the **Command** input window.



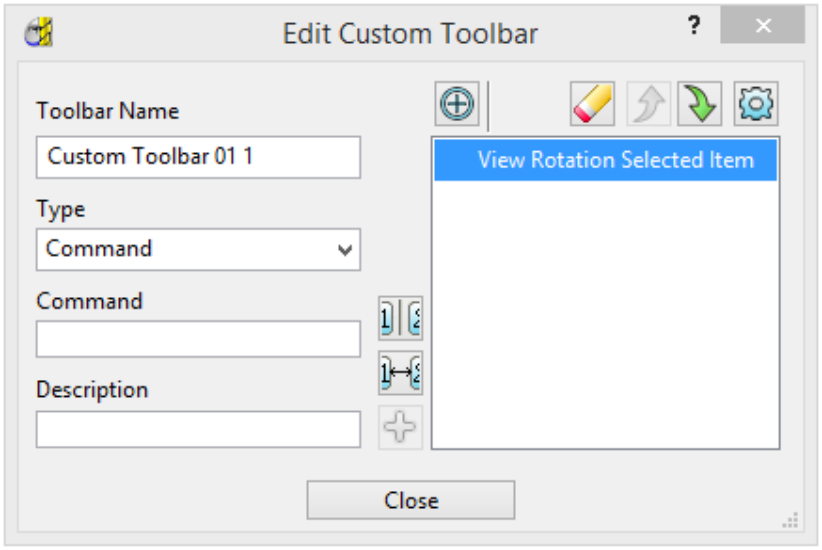


7 Drag the cursor across the required text (**VIEW CENTRE SELECTED**), right mouse click over it, and from the local menu select ***Copy***.

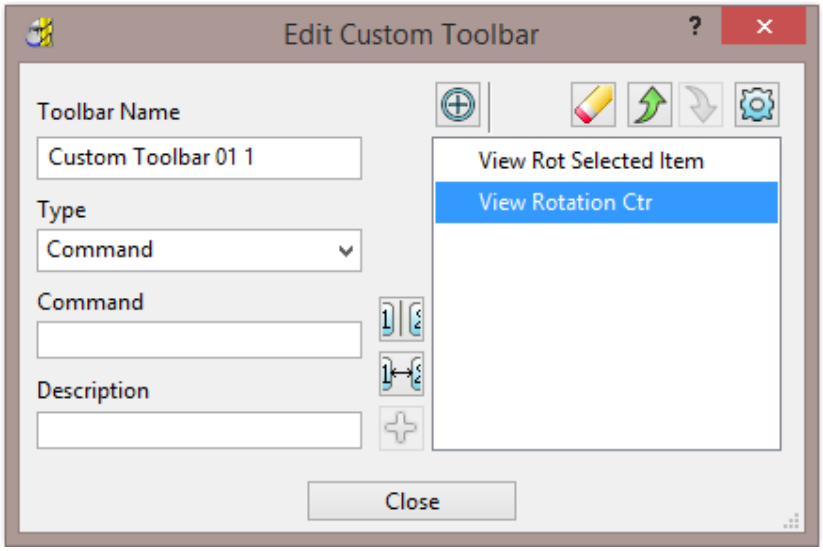
8 Back in the **Edit Custom Toolbar** dialogue paste the above text into the ***Command*** field.



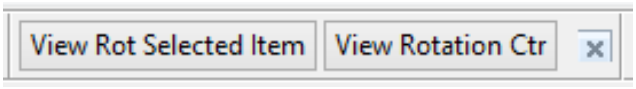
9 Input a suitable name in the ***Description*** field before selecting **Add selected item**. 



10 Repeat the above process to create a new toolbar option to restore the **View - Rotation Anchor** to use the **View** centre (equivalent text input = **VIEW CENTRE VIEW**).



The new custom toolbar (In the **PowerMILL** graphics area) should then look something like this:-



4. Tool Database

Introduction

The ***PowerMILL*** **Tool Database** enables the user to ***store*** and ***load*** pre-defined **Tools** in a single database. A **Tool** stored in the database can include definition for the **Shank**, **Holder**, **Speeds**, **Feeds**, **Stepovers**, and **Stepdowns**. It is also possible to assign different **Speeds** and **Feeds** to the same **Tool** where it is assigned with alternative **Stock Material** types.

The following chapter explains how to use the **Tool Database**. It is your responsibility to ensure that values selected from the database are suitable for purpose. If in doubt, relevant advice can be obtained from specialist tooling suppliers.

Accessing an existing Tool Database

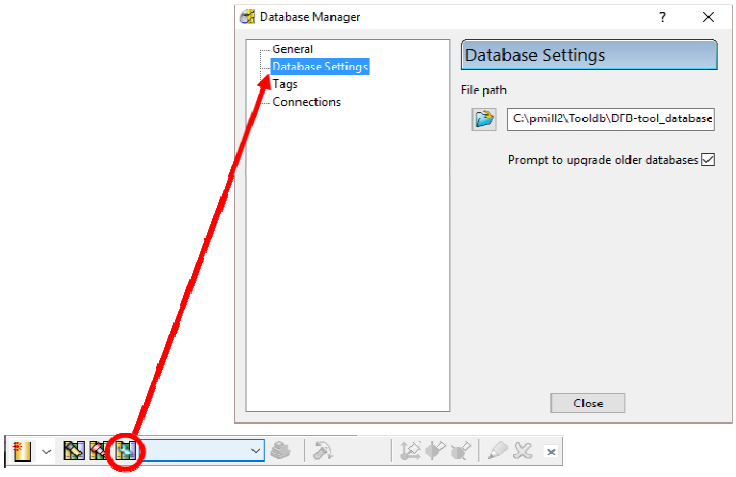
The first step in the process is to register the location of the **Tool Database** to be used in **PowerMILL**. In this example we will use a pre-defined database located in:- **….\pmill2**

1 In the **PowerMILL** ***explorer*** right mouse click on **Tools** and from the local menu select, **Toolbar**.

2 From the **Tool** toolbar select the ***Database Manager*** dialog. 

3 In the selected **Database** dialog select  and browse to the file:

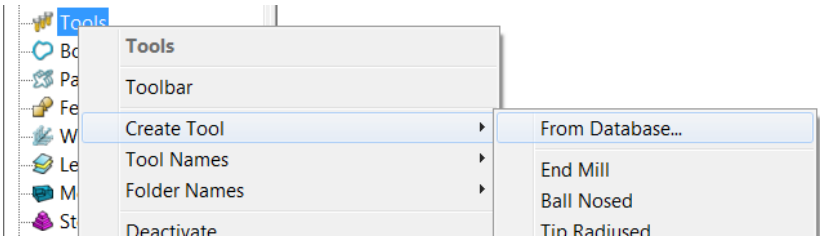
....\pmill2\Tooldb\Example-tool_database.mdb



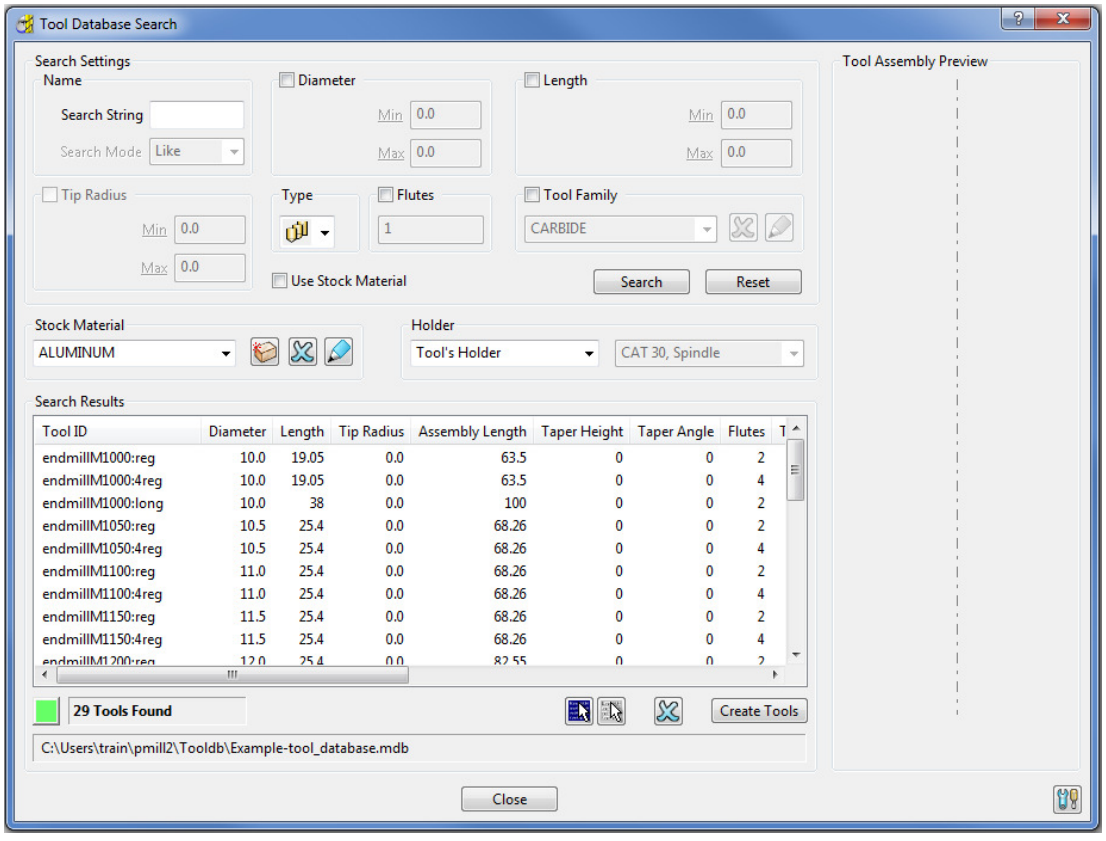
4 Close the **Database Manager** dialog.

Loading Tools from the registered Database

5 In the **PowerMILL *explorer***, right mouse click on **Tools** and select **Create Tool from Database**.



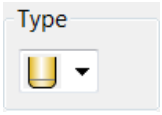
The **Tool Database Search** form will display all **Tools** stored in the currently registered **Tool Database** (Database path is displayed at the bottom of the dialog).



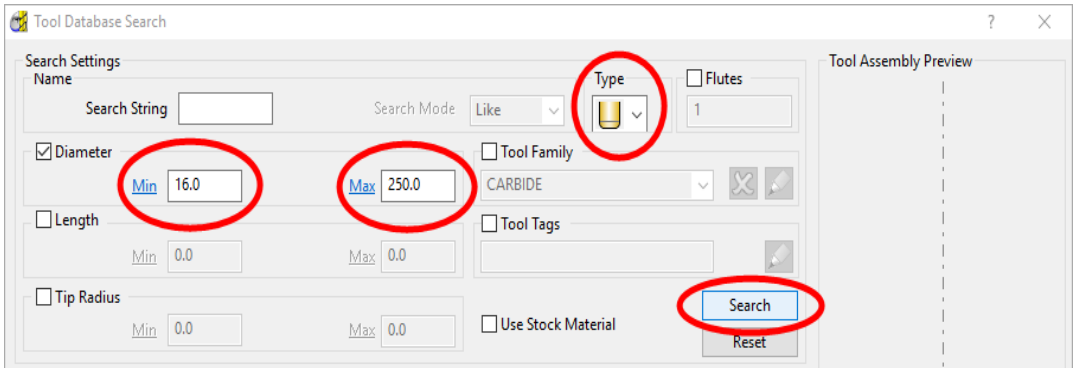
*Any displayed **Tool** can be loaded into the **PowerMILL** session, by double clicking on its name in the list.*

*To perform a more specific **Search**, reduce the options by selecting such criteria as the **Tool Type**, **Stock Material**, **Diameter range**, **Length range**, **etc** and select the **Search** button.*

6 Select the *Tool* **Type** option as **Tip Radiused**.

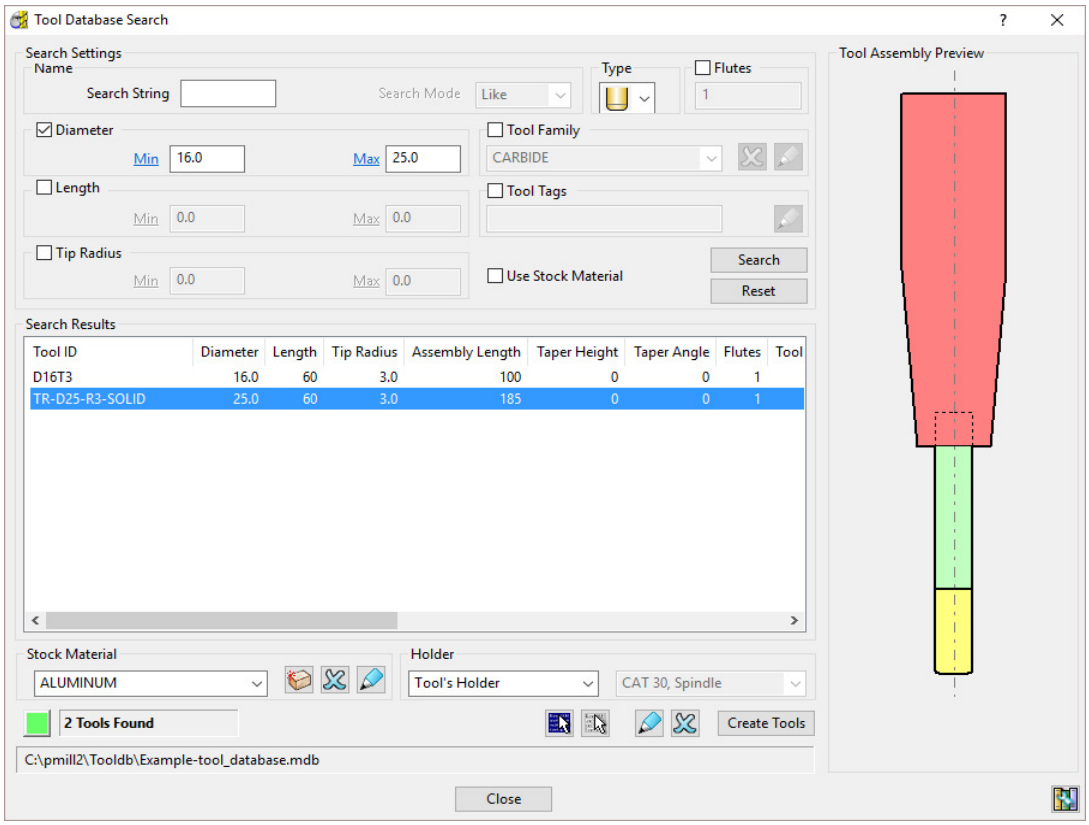


7 Tick the box named **Diameter** and input a range of **Min 16** and **Max 25**.



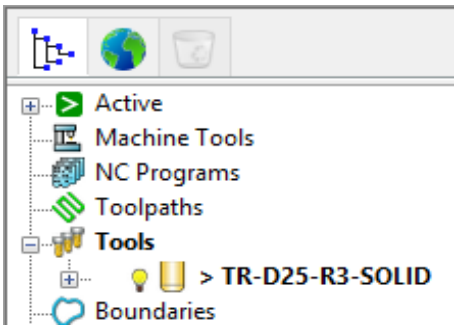
8 Click on the **Search** button to display the filtered down choice of **Tools**.

*All available **Tip Radiused tools** will be listed within the specified **Diameter range** - (2 Tools found).*



9 Double left mouse click on the displayed **Tool** (***TR-D25-R3-SOLID***) in the ***Search Results*** area to load it in to the **PowerMILL Project**.

10 Close the **Tool Database Search** dialog.

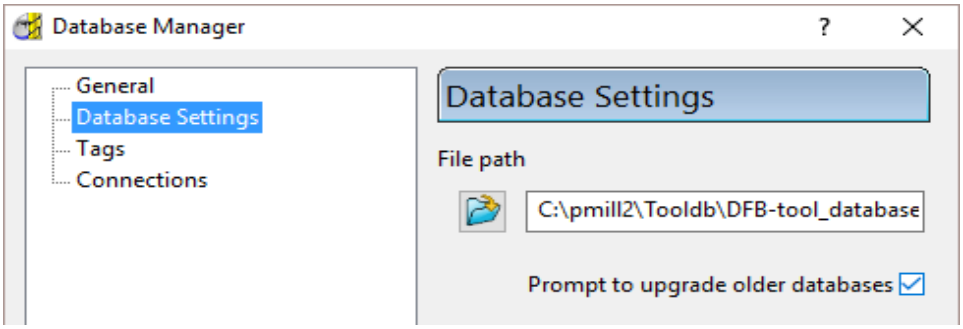


Exercise

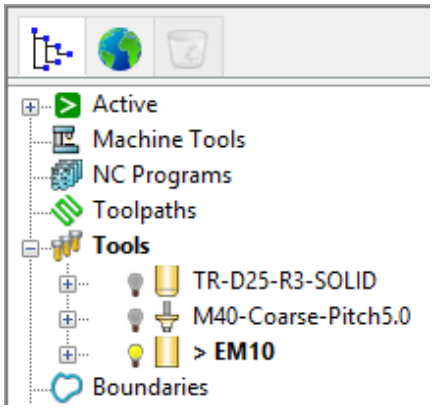
11 Do not delete the existing **Tool *TR-D25-R3-SOLID*** from the current **Project**.

12 Switch to a different **Tool Database**:- **….\pmill2\Tooldb\dfb-tool_database.mdb**

*To switch to the new **Database** select*  *from the **Tool** toolbar to open the **Database Manager** and browse the new **File Path***  *to the required **Tool Database**.*



13 Load in the ***Routing Tool*** - **M40-Coarse-Pitch5.0** and the ***End Mill*** - **EM10**.



*Do not delete the above **Tools** from this **PowerMILL Project** as they are required in the next section.*

Creating a new Tool Database

A copy of an empty **Tool Database** is stored within the installed **PowerMILL** software:-

C:\Program Files\Delcam\PowerMILL 20.0.10\file\tooldb\tool_database.mbd

This can be copied to a more accessible location such as:-

….\pmill2\Tooldb\tool_database.mbd

The copied file can be re-named as required, eg:-

My_tool_database.mbd

In the following tutorial a copy of the supplied (empty) **Tool Database** has already been copied and re-named to:-

….\pmill2\tooldb\tool_database.mbd

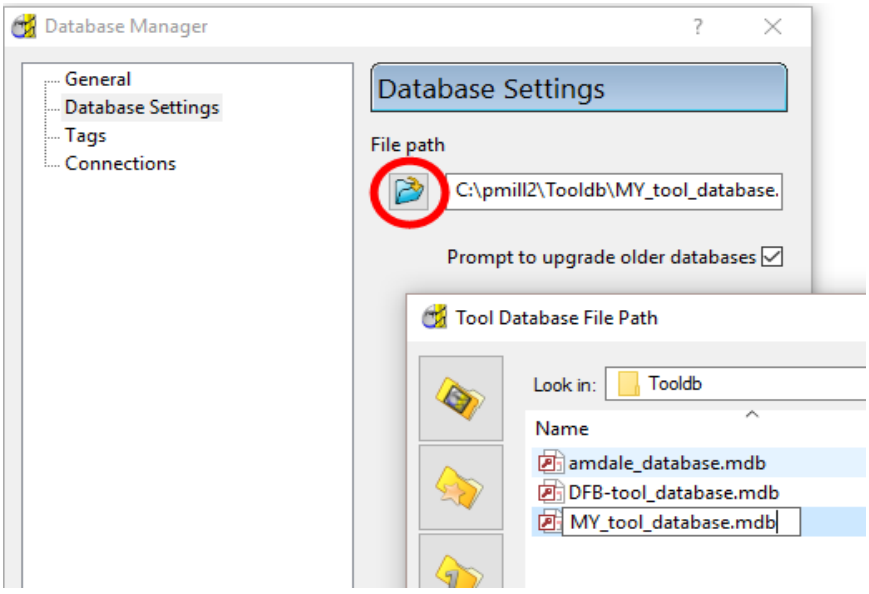
1 From the **Desktop** select **Windows Explorer** and in the dialog browse to the folder:-

....\pmill2\tooldb

2 Make a copy of the file **tool_database.mbd** and re-name it as:-

My_tool_database.mbd

3 From the **Tool** toolbar select **Database Manager** and in the dialog, browse to the new **Tool Database** to access the following page:-



4 **Open** the **Tool Database File Path** browser  and locate the file:-

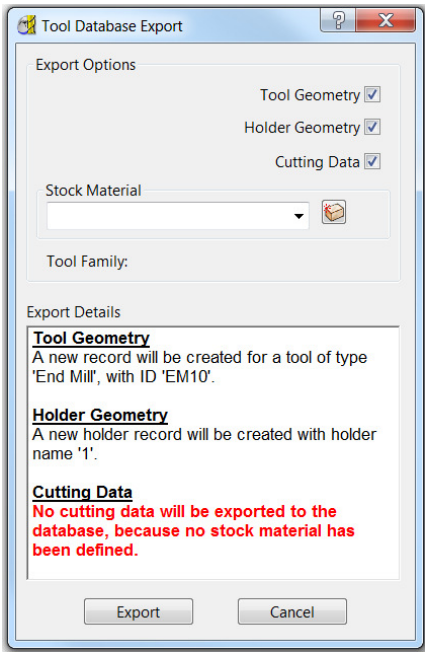
....\pmill2\Tooldb\My_tool_database.mbd

5 Select **Open** to register the required ***Tool Database*** to the **PowerMILL** session.

6 **Accept** the ***Options*** dialog.

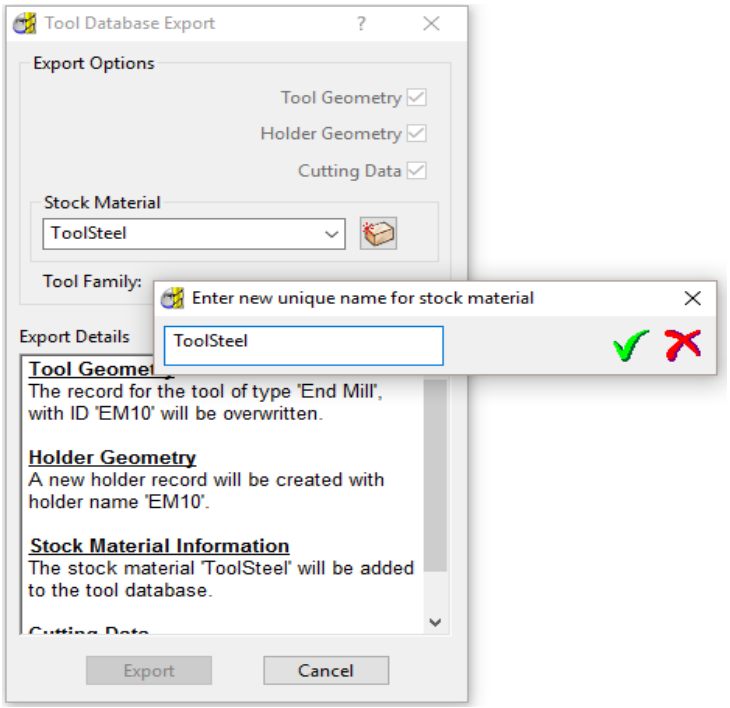
Adding tools to a PowerMILL Tool Database

7 In the **PowerMILL *explorer*** right mouse click on the tool **EM10** and from the local menu select **Add Tool to Database**.



*If there is a problem a warning message will be displayed in red text in the **Export Details** window.*

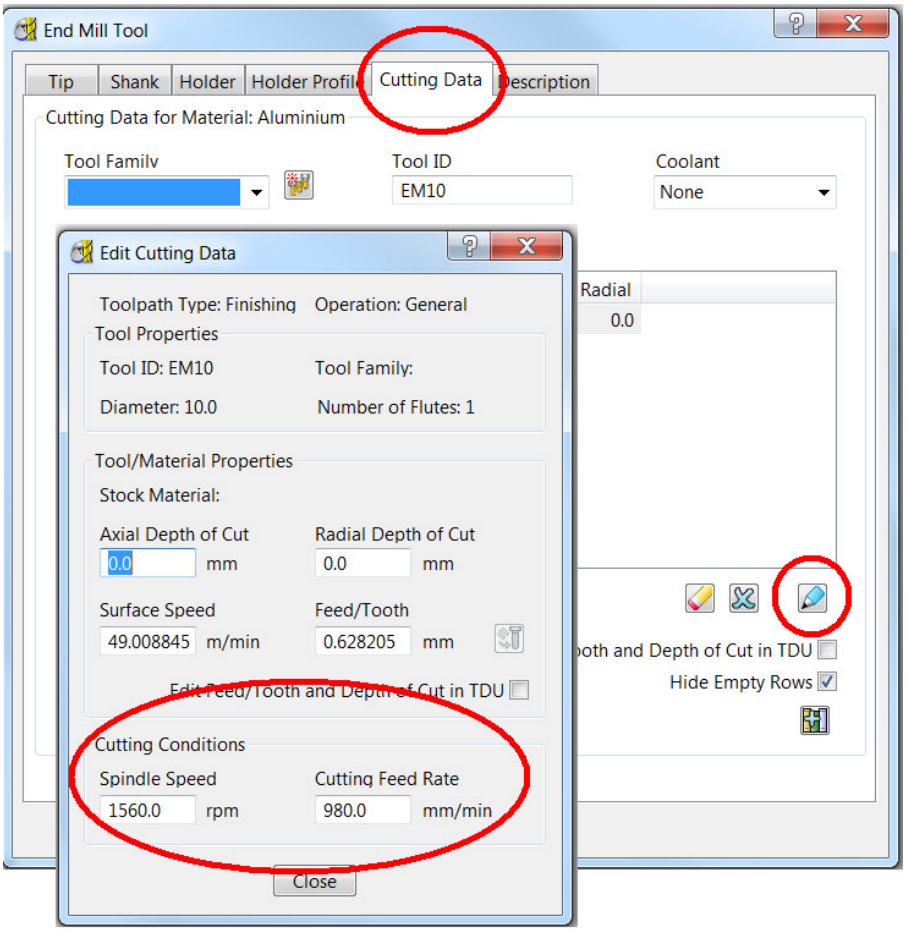
8 Select to open the **Enter stock material name** dialog and input the text **ToolSteel** and click on the ***green tick*** to accept.



9 Select **Export** and the ***End Mill* EM10** will be saved (along with the stored **Cutting Data**) in the **Tool Database** group for the ***Stock Material*** - **Tool Steel**.

Adding Cutting Data for different Stock Materials

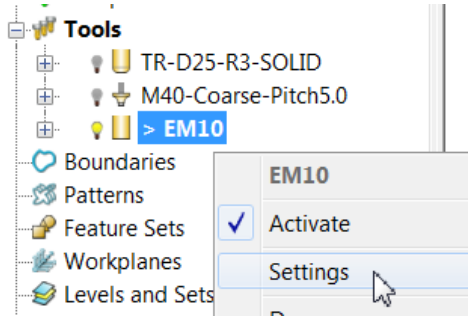
In **PowerMILL** suitable **Cutting Data** (Feeds, Speeds, Stepovers, etc) can be assigned to an individual tool.



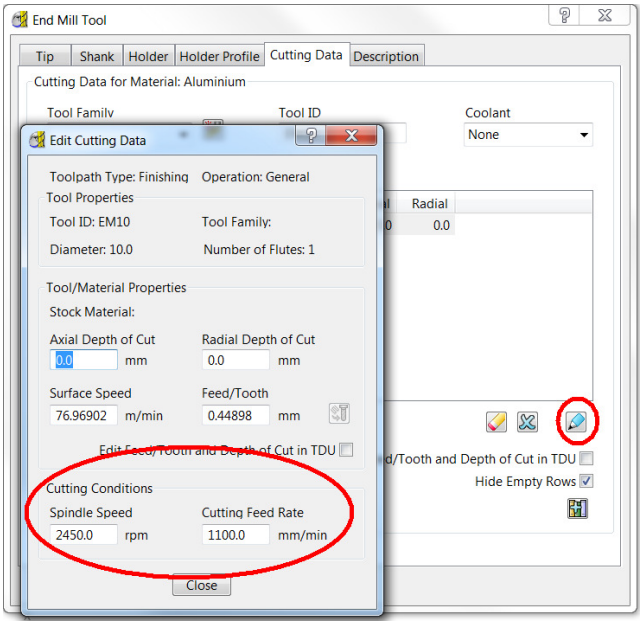
In this case the **Spindle Speed 1560** and **Cutting Feed Rate 980** are stored in the **EM10** tool definition.

In the **PowerMILL Tool Database** different **Cutting Data** can also be set up for different materials (Tool Steel, Aluminium, Graphite, etc) on an individual **Tool**.

10 In the **PowerMILL *explorer*** right mouse click on the ***Tool*** **EM10** and from the local menu select **Settings**.



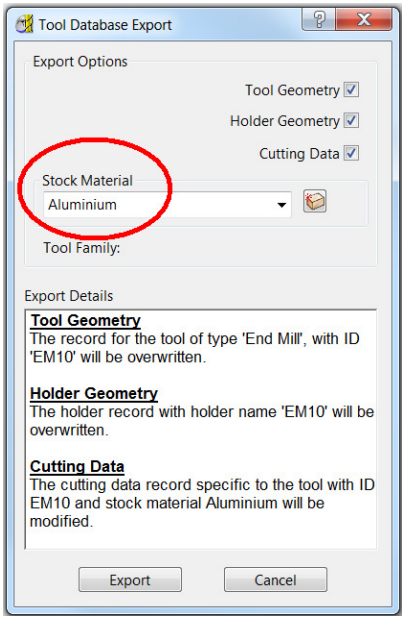
11 In the ***End Mill*** **Tool** dialog select the **Cutting Data** page.



12 Select the **Edit the cutting data** option. 

13 Change the **Spindle Speed** to **2450** and **Cutting Feed Rate** to **1100**.

14 In the **PowerMILL** ***explorer*** right mouse click on the ***Tool*** **EM10** and from the local menu select **Add tool to Database**.



15 Set **Stock Material** to **Aluminium** and then click on **Export**.

16 Edit the **EM10 Tool**, to have a **Spindle Speed** of **2880** and **Cutting Feed Rate** of **1650**.

17 **Export** the **EM10 Tool** to the **Tool Database** again but this time with the **Stock Material** set to **ToolSteel**.

When the **EM10** tool is created from the ***Tool Database*** the stored **Spindle Speed** and **Cutting Feed Rate** will be different for an **Aluminium** or **ToolSteel**, ***Stock Material*** selection.

5. Interactive Macros

Introduction

In this chapter selected, existing **macros** will be enhanced to include more comprehensive, interactive control options such as; On Screen User Prompts, Error and Invalid Input Checking, Loops, etc.

Initial Macro file management

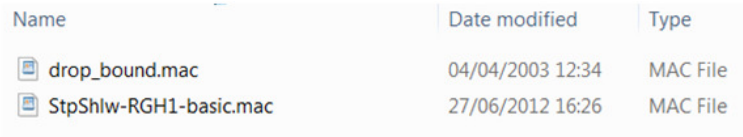
1 From the desktop, open a **Windows Explorer** 

dialog.

2 Browse to:-

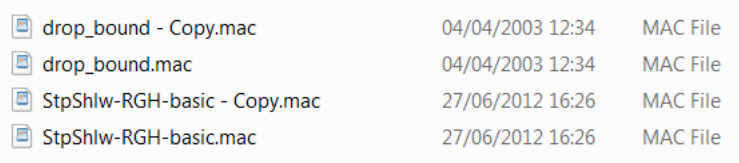
...\PowerMILL_Data\InteractiveMacroTraining

3 The folder contains the following 2 **PowerMILL macro** files.



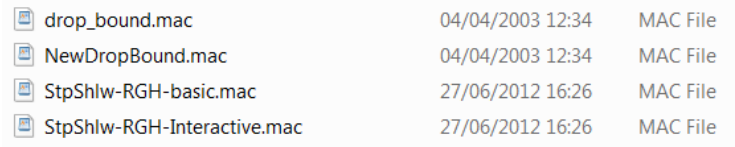
4 Make a copy of both macro files;

(Shift, select both files, right mouse click on them, and from the local menu select **Copy**, then right mouse click in the explorer window and from the local menu select **Paste**).



5 **Rename** the 2 new files as:-

NewDropBound.mac and **StpShIw-RGH -Interactive.mac**



Drop Boundary Macro

Development of an ***interactive macro*** starting from a basic ***Drop Boundary*** macro:-

- Setting up a **Boolean** variable.
- Use of the **IF** statement to trap an error condition.
- Input suitable commands that only allow the **macro** to continue if a specified entities are **Active** (***Boundary*** and ***Tool***).
- Locate the correct syntax for parameter identifiers (**Help** - **Parameters** - **Summary**).
- Input suitable commands that only allow the **macro** to continue if a **Block** exists (Assumes the **Block** is valid if it has a Z height (Zmax–Zmin = 0)).

Setting up ready for the DropBound.mac enhancements

1. From the ***main*** pull down menus, select **File** - **Import Model**:-

   ...\PowerMILL_Data\models\Zlimits.dgk

2. Select **File** - **Save Project As**:-

   ...\Coursework\PowerMILL_Projects\MacroTests

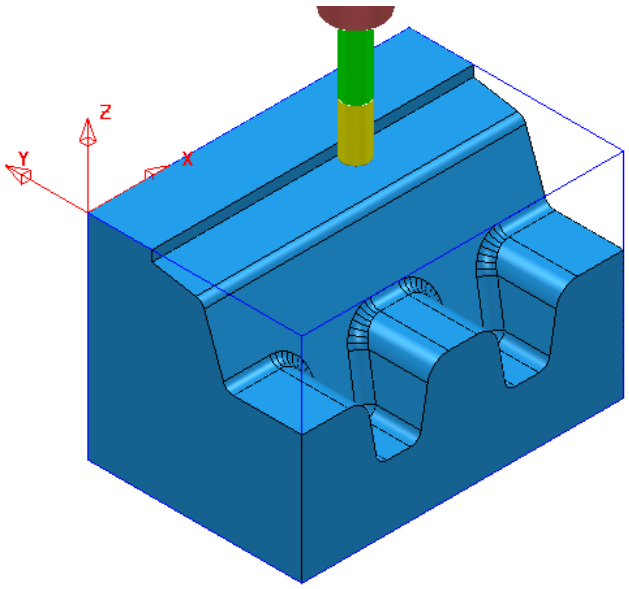
3. Select an **ISO 1** view and **Activate** the imported **Workplane** 1.

4. From the local ***user menu*** import the '***TipRad***' tool **D12TR1**:-

   (**Tip** Length 25 -- **Shank** Length 40 –- **Holder1** Dia (Lower 25 Upper 40) –- **Holder2** Dia 40).

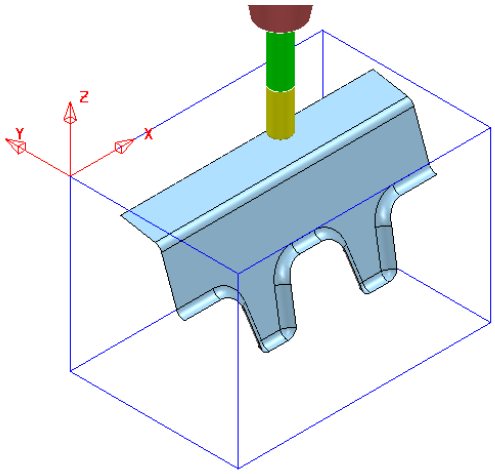
5. **Calculate** a **Block** using the default settings.

6. **Calculate** the **Rapid Move Heights** using the default settings.

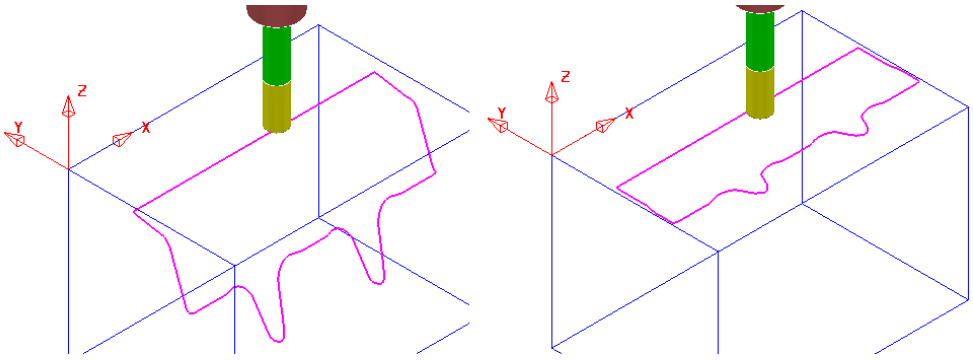


Creation of NewDropBound.mac enhancements

1 Back in **PowerMILL** create a default ***Selected Surface*** **Boundary** using the following faces of the model.



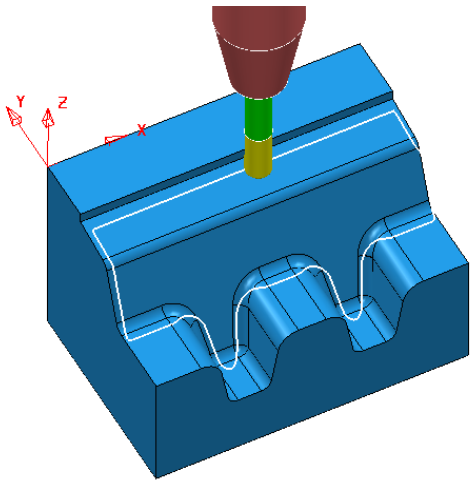
2 Right mouse click on the **Boundary** and from the local menu select **Edit** – **Flatten**.



3 From the main pull down menus select **Tools** – **Customise Paths** and in the dialog browse to:-

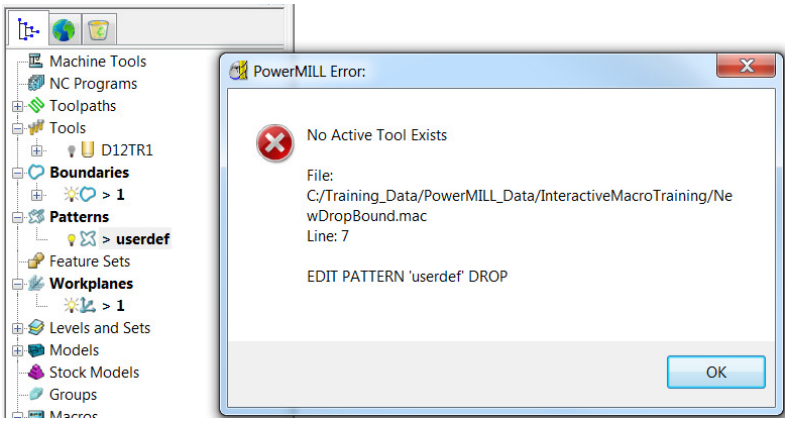
...\PowerMILL_Data\InteractiveMacroTraining

4 From the **PowerMILL** ***explorer*** select **Macros** – **InteractiveMacroTraining** and from the list **Run** the macro **NewDropBound**.



5 Right mouse click on the **Boundary** and from the local menu select **Edit** – **Flatten**.

6 De-Activate the **Tool *D12TR1*** and then **Run** the macro **NewDropBound**.



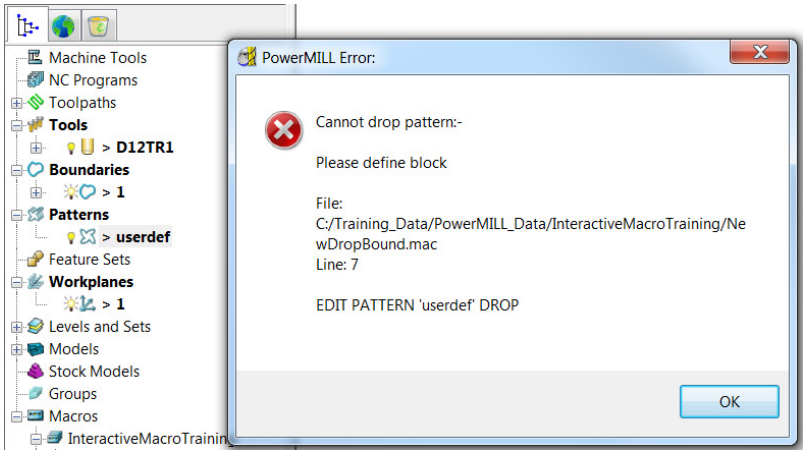
A standard **PowerMILL Error** dialog is opened reporting that ***No Active Tool Exists***. The original **Boundary** is un-changed and the processing has not prevented the creation of an unwanted, empty **Pattern *userdef***.

7 In the **PowerMILL Error** dialog select **OK** to close.

8 **Delete** the **Pattern *userdef***.

9 **Activate** the **Tool *D12TR1*** and in the **Block** dialog, delete all dimensions  .

10 From the **PowerMILL *explorer*** select **Macros** – **InteractiveMacroTraining** and from the list **Run** the macro **NewDropBound**.



A standard **PowerMILL Error** dialog is opened reporting that it ***Cannot drop pattern*** – ***Please define block***. The original **Boundary** is un-changed but as before the processing has not prevented the creation of an unwanted, empty **Pattern *userdef***.

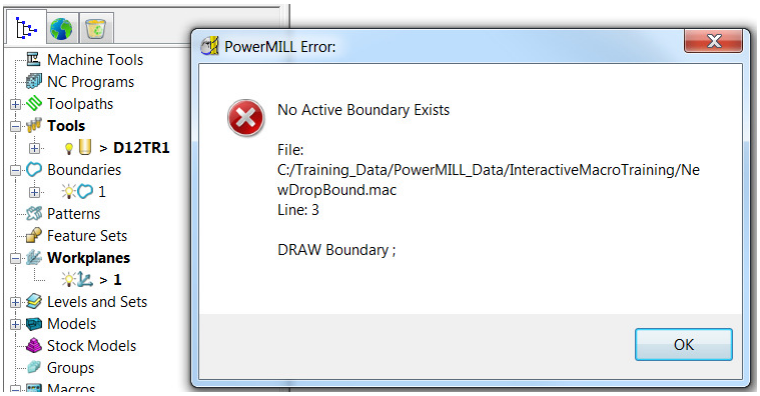
11 In the **PowerMILL Error** dialog select **OK** to close.

12 **Delete** the **Pattern *userdef***.

13 **Activate** the **Tool *D12TR1***.

14 In the **Block** dialog select **Calculate** followed by **Accept**.

15 From the **PowerMILL *explorer*** select **Macros** – **InteractiveMacroTraining** and from the list **Run** the macro **NewDropBound**.



If the **Boundary** is not **Active**, the **Tool** is **Active**, and the **Block** correctly defined, further processing is impossible and the empty **Pattern *userdef*** will not be created.

16 In the **PowerMILL Error** dialog select **OK** to close.

As demonstrated this **macro** will only work correctly if certain conditions apply (***Active Boundary*** - ***Active Tool*** – ***Defined Block***). If these conditions apply the **macro** will work (even if the empty **Pattern *userdef*** still exists). It would however be more efficient to in-build some preliminary checks into the macro to both advise the user and stop the macro from processing.

17 In the **Windows Explorer** dialog open the file **NewDropBound.mac** in either **Wordpad** or **Notepad**.

```
//
//
DRAW Boundary ;
EXPLORER SELECT Boundary ; NEW
EXPLORER SELECT Pattern 'userdef' NEW
EDIT PATTERN 'userdef' INSERT BOUNDARY ;
EDIT PATTERN 'userdef' DROP
DELETE BOUNDARY ;
CREATE BOUNDARY ; SKETCH FORM BOUNDARY
EDIT BOUNDARY PREINSERT PATTERN "userdef"
EDIT BOUNDARY ; INSERT PATTERN FROM_COMBO
BOUNDARY ACCEPT
EXPLORER SELECT Pattern 'userdef' NEW
DELETE PATTERN 'userdef'
REFRESH VIEWMILL RESIZEVIEW
//
//
```

18 Input a **Title** and **Description** at the start of the **macro *NewdropBound***.

*Two forward slashes **//** form a blank line and are used for gaps or text only lines. Note: A real blank line can also be used as a gap in the macro but obviously will not support text comments.*

```
// [-----------------------------------]
// [********** NewDropBound.mac **********]
// [-----------------------------------]
//
// ********************************************************
// *** This Macro will Drop the ACTIVE BOUNDARY onto the  ***
// *** MODEL relative to the (Finished Size) tip position ***
// *** for the ACTIVE TOOL ***
// ********************************************************
//
```

19 Next add the following combination of comment and command lines to check that an **Active Boundary** exists (***Command lines*** displayed in red text)).

```
//
// **** Check that an Active Boundary exists ****
//
// Setup a default Boolean variable to trap the possible error of no Active Boundary
// A Boolean value can either = 0(false) or 1(true)
   bool $ActiveBoundCheck = 0
// Now use the ERROR boolean function to check if an Active Boundary exists.
// Test this in PowerMILL by typing: PRINT = ERROR entity('boundary','').name
   $ActiveBoundCheck = ERROR entity('boundary','').name
// If $ActiveBoundCheck returns TRUE, then there is no Active Boundary.
// We will print an error message and exit the macro.
   IF ($ActiveBoundCheck) {
   message error "There is no Active Boundary. Activate a Boundary and try again!"
   return
  }
//
```

20 Repeat the above command lines replacing the variable name **$ActiveBoundCheck** with **$ActiveToolCheck** and entity **boundary** with **tool**.

```
//
// **** Check that an Active Tool exists ****
//
   bool $ActiveToolCheck = 0
   $ActiveToolCheck = ERROR entity('tool','').name
   IF ($ActiveToolCheck) {
   message error "There is no Active Tool. Activate a Tool and try again!"
   return
  }
//
```

Exercise

21 Next create a test for a valid **Block** by using the situation of **Max Z** and **Min Z** values being set to 0. This can be used to trap the error in an **IF** statement. The format is similar to the previous two checking actions apart from the calculation line to trap a value for the ***Boolean***, test.

a Use $CalculatedBlockCheck as the ***Boolean variable*** name (set to false).

b You need to trap a value into $CalculatedBlockCheck from the **Block** dialog using the calculation **Zmax**-**Zmin**=**0** for an error condition.

From the main pull down menus select **Help** – **Parameters** – **Summary** and scroll down the HTML page to locate the correct parameter identifier:-

Eg. For the **Zmax** value the correct parameter identifier is Block.Limits.ZMax.

In the macro replace **Zmax** and **Zmin** (shown below) with the correct syntax strings as located in **Help** – **Parameters** – **Summary**.

```
$CalculatedBlockCheck = (( Zmax – Zmin)  == 0)
```

c Input the **IF** statement lines:-

```
//
IF ($CalculatedBlockCheck) {
   message error "There is no defined Block. Create a Block and try again!"
   return
  }
//
```

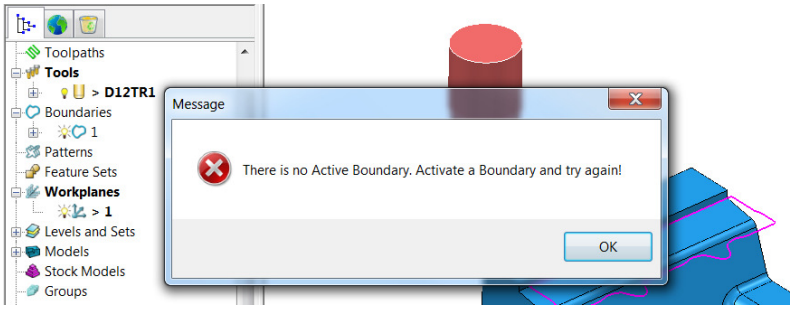
End of Exercise section!

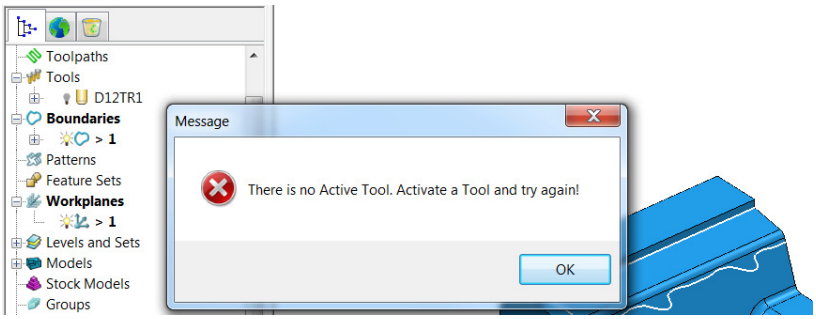
22 A further enhancement to the macro would be to ensure that the resultant dropped Boundary is calculated to fixed finished **Thickness** and **Tolerance** values.

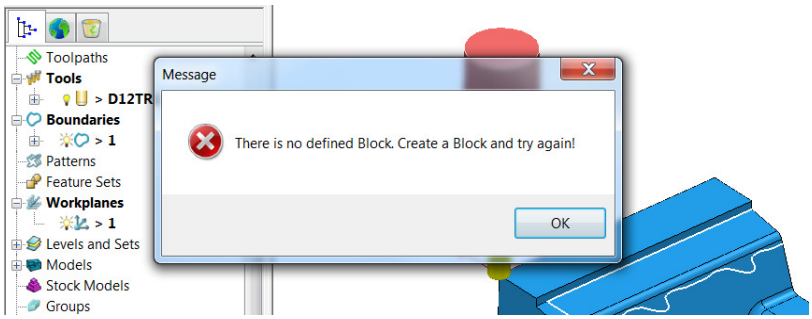
23 Insert the following comment and command lines into the macro immediately after the 3 boolean checks:-

```
   message error "There is no defined Block. Create a Block and try again!"
   return
  }
//
// *** Ensure that the Finishing Tolerance and Thickness are set ***
EDIT PAR 'Tolerance' "0.01"
EDIT PAR 'Thickness' "0"
//
   DRAW Boundary ;
   EXPLORER SELECT Boundary ; NEW
   EXPLORER SELECT Pattern 'userdef' NEW
```

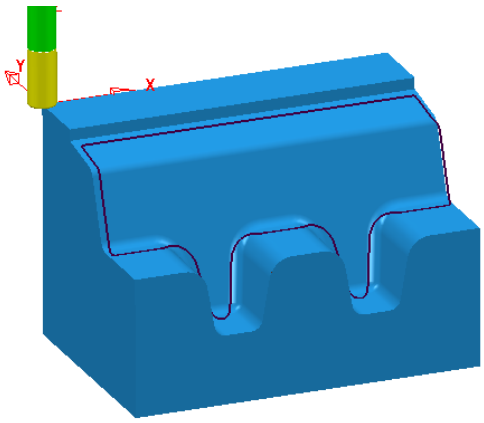
24 Thoroughly test the new features in the **macro** by re-running with no **Active *tool***, no **Active *boundary***, no defined **Block**.







25 Then try the **macro** with all valid conditions active.



26 Delete all **Boundaries** from the **Project**.

27 Select **File** - **Save Project** to update the stored entities.

28 Do not close the **Project** as it will be used in the next example.

Steep and Shallow Roughing Macro

The development of an ***interactive macro*** that uses the **Steep and Shallow Finishing** strategy to produce an alternative 3D Roughing process.

Stage -1

- Setting up a **Boolean** variable.
- Use of the **INPUT** option to prompt the user to enter a ***Maximum Thickness*** value.
- Use of **ELSE** and **ELSEIF** statements to filter out invalid input to the ***Maximum Thickness*** value.
- Use of **DO** - **WHILE** loop option that returns the process back to the prompt line to enter a new ***Maximum Thickness*** if the specified value is invalid.

Stage -2

- Addition of a **DO** - **WHILE** loop that will continually produce **Steep and Shallow Finishing** toolpaths. The loop will continue while the progressively, decreasing **Thickness** value is greater than **0.5**.
- Addition of an **IF** - **NOT** command to allow the macro to recognise and delete an un-calculated toolpath (**Thickness** value too large for the **Toolpath** to fit within the **Block** dimensions).
- Finally each toolpath will be ***collision checked*** using a **Verification Thickness** value of **0.5**.

Development of the StpShlw-RGH-Interactive.mac

Stage 1:-

1 In the **Windows Explorer** dialog open the file; **StpShlw-RGH-Interactive.mac** in either **Wordpad** or **Notepad**.

```
// *** Set up the parameter $thic to use Real values ***
  real $thic = 0
  $thic = INPUT "Thickness"
//
//
IMPORT TEMPLATE ENTITY TOOLPATH TMPLTSELECTORGUI
"Finishing\Steep-and-Shallow-Finishing.002.ptf"
EDIT PAR 'InterConstzOrder' 'combined'
EDIT PAR 'ThresholdAngle' "5"
EDIT PAR 'Overlap' "0"
EDIT PAR 'CutDirection' 'climb'
EDIT PAR 'ShallowCutDirection' 'climb'
EDIT PAR 'RadialDepthOfCut.UserDefined' '1' Edit Par 'Stepover' "3"
EDIT PAR 'AxialDepthOfCut.UserDefined' '1' EDIT PAR 'ShallowStepover.Value' "3"
```

```
EDIT PAR 'ShallowStrategy' 'raster'
EDIT TPPAGE RASTERPAGE
EDIT TPPAGE STEEPSHALLOW
EDIT PAR 'Tolerance' "0.1"
EDIT PAR 'Thickness' $thic
// EDIT TOOLPATH ; REAPPLYFROMGUI
EDIT TOOLPATH ; CALCULATE
SteepShallowFin ACCEPT
DRAW TOOLPATH ;
//
//
```

The start of this **macro** creates a variable (**$thic**) that will accept **Real** numbers, and then assigned with an initial value of **0**.

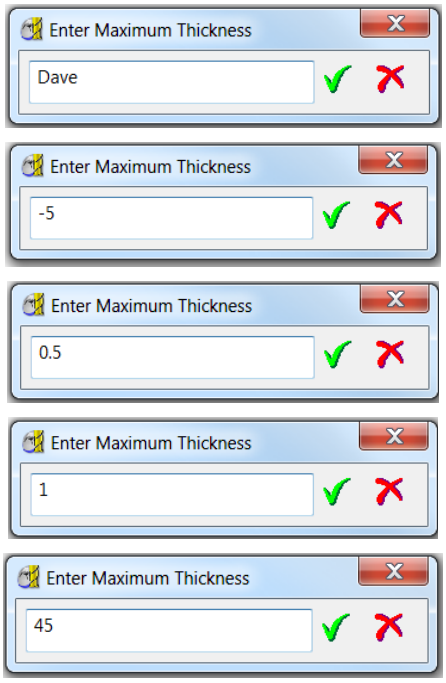
The first task is to create a **Boolean variable** to both prompt the user to enter a value and to test for a valid input.

2 Input the following lines after; real $thic = 0 (Note: ***command lines*** are displayed in Red text).

```
// b
// Create a boolean variable which will be used with the "ERROR" function to check the users response
  bool $err = 0
   do {
   $thic = INPUT "Enter Maximum Thickness"
// Check the user entered a real number. $err will be FALSE if the value is good.
   $err = ERROR $thic
// If the users did not enter a good value, print an information message explaining the error.
   if $err {
   message error "Thickness Value must be numerical. Please try again."
// If the user entered a real number less than or equal to 0.5, print an error message.
 } elseif ($thic <= 0.5) {
   message error "Value must be greater than 0.5 --- Please try again."
 }
// If the user entered an incorrect value, go back to the "do" command (Ask for Thickness again!)
  } while ($err or $thic <= 0.5)
//
```


3 **Save** the file, but do not close it down as further commands will be added in 'Stage 2' after testing the latest additions.

4 Thoroughly test the new features of the **macro** by entering the following invalid inputs at the user prompt stage:-



*Only the last 2 user inputs are valid, allowing the **macro** to continue processing beyond this stage.*

The ***Maximum Thickness*** value **1** entered above is the only user input that produces a physical **Toolpath**.

When the valid, input value **45** is entered above, an uncalculated **Toolpath** appears in the **PowerMILL *explorer***. This is due to **Thickness** value of **45** being fully outside the **Block** dimensions relative to the **Model** form.

In 'stage 2' some **command lines** will be added that will automatically delete any resultant, uncalculated **Toolpaths**.

5 Assuming all the above checks out, you are now ready to proceed to Stage 2.

Stage 2:-

A second ***Boolean variable*** (**$count**) will be added at the start of the **macro** that will only accept **Integer** values. It will be assigned with a default value of **0**.

6 Input the following line int $count = 0 between the existing lines:-

```
bool $err = 0
int $count = 0
do {
```

*The default value **Boolean variable** value **0** must be input before the start of the do loop.*

7 Just after the **do – while** options and immediately before the **Steep and Shallow**, set up and machining commands insert the following:-

```
//
$Count = $thic/2
While Count > 0 {
//
```

The **$Count**, integer value is based on the **Maximum Thickness** value divided by 2. This will allow for the **Thickness** value being progressively reduced by 2 for each subsequent toolpath.

8 Immediately following the **Steep and Shallow**, set up and machining commands insert the following:-

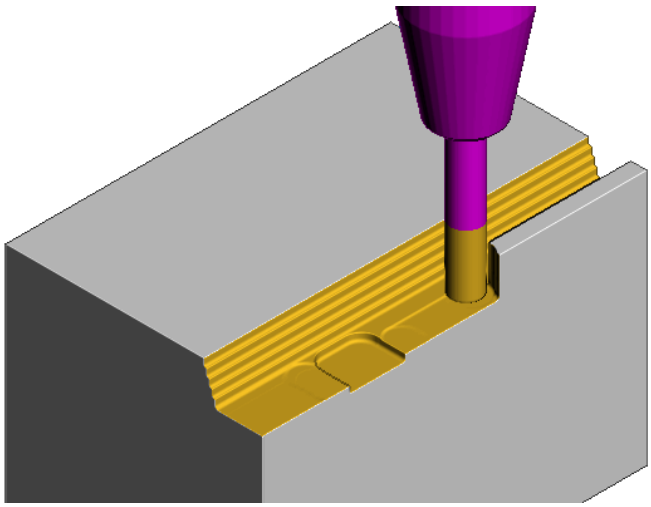
```
//
$Count = Count – 1
$thic - $thic – 2
}
//
```

The current value stored in **$Count**, is reduced by 1 and **$thic** by 2 ready for the return to the start of the **do** loop (}).

9 **Save** the file, but do not close it down as further commands will be added in after testing the latest additions.

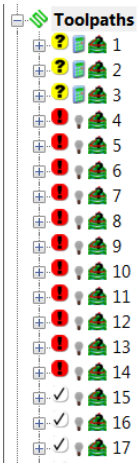
10 Test the new features of the **macro** by entering 24 as the ***Maximum Thickness*** value at the user prompt stage:-

It will be noted that the first **Toolpath** listed in the **PowerMILL *explorer*** is taking a heavy cut into the material block. The **Maximum Thickness** prompt needs a higher value to allow for the maximum distance between the **model** and the **block**.

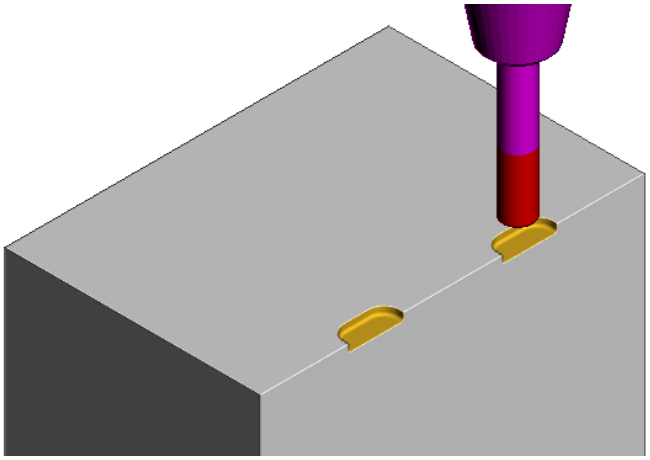


11 **Delete** all **toolpaths** and re-run the **macro** but this time with 46 as the ***Maximum Thickness*** value at the user prompt stage:-

It will be noted that 3 un-processed **Toolpaths** exist at the start of the list in the **PowerMILL *explorer***.



However the first of the ***calculated*** toolpaths (**4**) in the list is applying the required depth of cut of around 2mm.



If a **toolpath** cannot be calculated (due to being outside the **Block** limits) then the **macro** can be set up to delete it from within the **DO** – **WHILE** loop.

29 At the end of the current **macro**, just before the updates to $Count and $thic add the following command lines:-

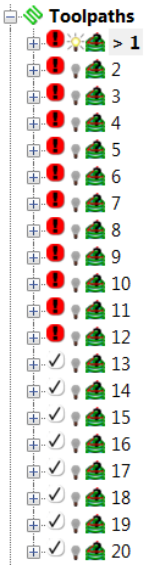
```
//
IF NOT computed {
Delete Toolpath ;
}
//
//
$Count = Count – 1
$thic - $thic – 2
 }
 //
```

30 **Save** the file, but do not close it down as further commands will be added in after testing the latest additions.

31 Delete all **Toolpaths** in the **PowerMILL *explorer***.

32 Test the new features of the **macro** by entering **46** as the ***Maximum Thickness*** value at the user prompt stage:-

33 This time there are no un-calculated toolpaths in the **PowerMILL *explorer*** list. The first **Toolpath** in the list is applying the required 1st depth of cut of around 2mm (as shown in the previous image).



Most of the **Toolpaths** in the top half of the above list are displaying the red gouge/collision warning. This has occurred due to the high **Thickness v**alue used.

The final stage for the **Interactive** part of the **macro** is to apply a ***gouge check*** to all toolpaths using a **Verification Thickness** value of **0.5**.

This will be inserted at the end of the **DO** – **WHILE** loop just after the check and deletion operation for an un-calculated **Toolpath**.

34 Record a temporary **macro** containing the necessary commands to perform a ***gouge check*** that uses a separate **Verification Thickness**:-

35 Insert the contents of the temporary **macro** at the end of the main **macro**, just before the updates to **$Count** and **$thic**:-

```
EXPLORER SELECT Toolpath ; NEW
FORM COLLISION
EDIT COLLISION TYPE GOUGE
EDIT COLLISION STOCKMODEL_CHECK N
EDIT COLLISION SCOPE ALL
EDIT COLLISION SPLIT_TOOLPATH N
EDIT PAR 'Verification.UseVerificationThickness' 1
EDIT PAR 'Verification.Thickness' "0.5"
EDIT COLLISION APPLY
COLLISION ACCEPT
//
```



```
$Count = Count – 1
$thic - $thic – 2
 }
 //
```

The final addition to the ***interactive*** part of the **macro** is to stop **PowerMILL** from issuing ***messages*** while the **toolpath verification** process is running.

36 Insert the following lines (shown red) before and after the **toolpath verification** command section as shown below.

```
//
// *** Temporarily switch off messages ***
DIALOGS MESSAGE OFF
//
EXPLORER SELECT Toolpath ; NEW
FORM COLLISION
EDIT COLLISION TYPE GOUGE
EDIT COLLISION STOCKMODEL_CHECK N
EDIT COLLISION SCOPE ALL
EDIT COLLISION SPLIT_TOOLPATH N
EDIT PAR 'Verification.UseVerificationThickness' 1
EDIT PAR 'Verification.Thickness' "0.5"
EDIT COLLISION APPLY
COLLISION ACCEPT
//
// *** Switch on messages ***
DIALOGS MESSAGE ON
//
```

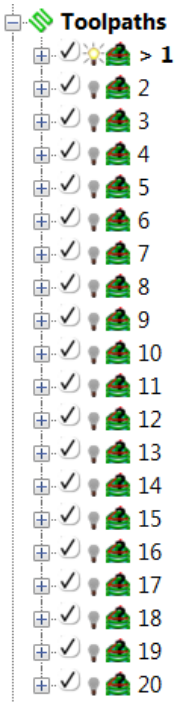
*Note it is good policy to include **comment lines** wherever appropriate to explain key actions in the macro (An example of one shown above:- // *** Switch on messages***).*

37 **Save** the file, but do not close it down as further commands will be added in after testing the latest additions.

38 Delete all **Toolpaths** in the **PowerMILL *explorer***.

39 Test the new features of the **macro** by entering **46** as the ***Maximum Thickness*** value at the user prompt stage:-

This time all toolpaths should display the ***tool, gouge safe*** symbol (black tick on white background).



Exercise

The **macro** is currently creating a series of toolpaths with progressively decreasing **Thickness** values that start at the user's max specified value reducing in steps of **2**.

40 **Copy** the original, **Steep and Shallow** finishing command lines from the macro and **Paste** back in after the **DO** – **WHILE** process.

No need to include the recently added toolpath verification command lines!

41 Change the new section to produce a final pass with a fixed **Thickness** value of **0.5**.

42 **Save** the file.

43 Delete all **Toolpaths** in the **PowerMILL *explorer***.

44 Test the new features of the **macro** by entering 46 as the ***Maximum Thickness*** value at the user prompt stage:-

The toolpaths produced should start from the initially ***entered*** max **Thickness** value of 46, progressively drop down in increments of 2 with the final passes being performed a 2 and 0.5 respectively.

*If an **odd** value for **max thickness** is entered then the final 2 passes will be performed at 3 and 0.5 respectively!*

End of Exercise section!

