

# SYNERGY HOLE FEATURES

This document is a training manual for hole features. It will explain to you what a hole feature is, how to use them in the system and, how to create custom hole features. The menu for hole features can be found in *Variational->Vmill->Holes* (See Figure 1.)

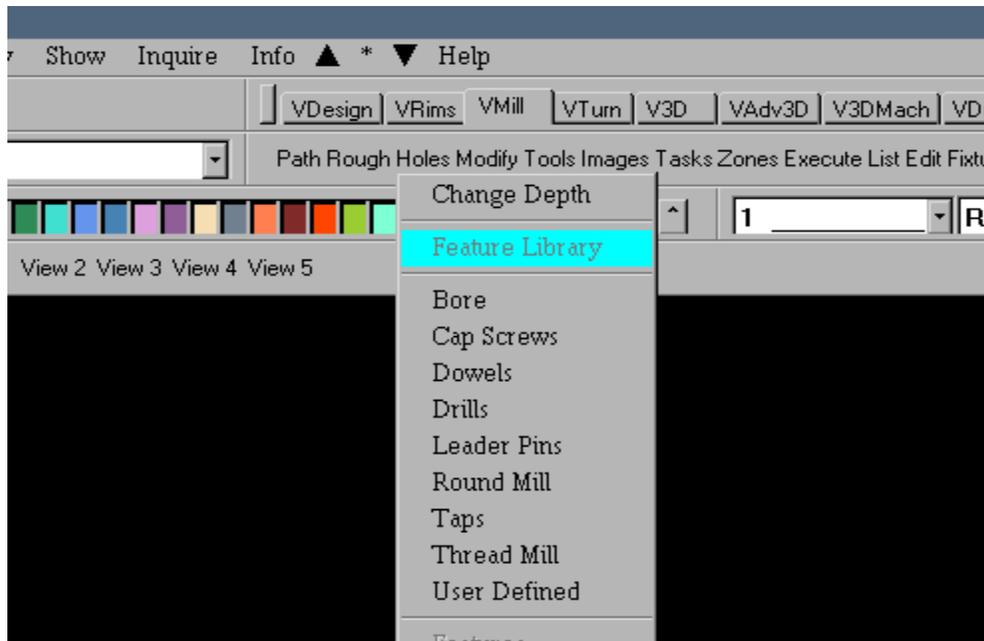


Figure 1.1 Hole Feature Menu.

## SECTION 1. Using Hole Features

### What is a hole feature?

A hole feature (or feature) is a specific type of Variational Strategy. A strategy is a play back method used in Variational Design (VDM) to capture many sets of data and apply them against existing Synergy geometry. A hole feature places one or more holes at a location. Each hole represents the application of a tool when manufacturing that hole location (See Figure 2.). To start, choose Drills from the Feature Menu.

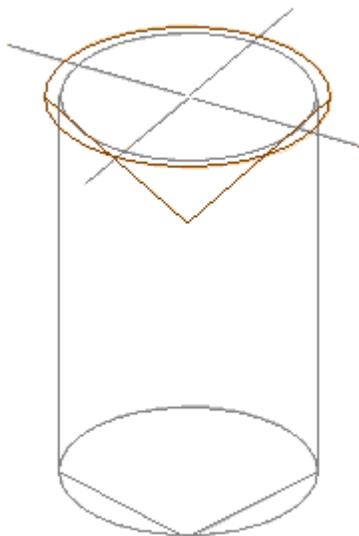


Figure 1.2 Drilled hole with spot and twist drills.

## Placing a hole feature

After picking the menu you will get a choice of the available hole features. For our examples choose the twistdr.250 from the available choices.

When you choose a hole feature you are asked to place it by choosing a group. This group can consist of either points or circles. Hole features will be placed at the point location and at the center of a circle. It will follow the Z location of the point or circle. To place a single hole feature, choose the Single option in the input line. This will temporarily take you out of the group mode and allow you to enter a location. Since hole features are variational, an element must already exist for you to pick. The single option is the repeat value on the input line. So, it can be chosen easily by center clicking the mouse if you are entering more than one location. A multiple option may be available in the future but was not available at the time of this document.

The New option on the input line will allow you to choose another hole feature of the same type without going back to the menu. If you want a different type of hole feature then you need to return to the menu.

On the hole features menu there is an option for multiple tools. This function can be used to choose a hole feature that has already been read into the system. That way a hole feature can be recalled for use at another location. This is used a lot when doing rotary work and using work zones. The hole features are placed in the same way as when reading in the hole feature. However, the New option at this point will bring up a list of previously loaded hole features.

## Changing the information

After placing a hole feature it is sometimes necessary to edit the information that defines it. The hole feature data is contained in a VDM record that cannot be picked on the screen. The hole feature process is done in two or more records. The hole feature data is contained in one record, and the placement of it is contained one or more records after it. You can see this by using the record list hotkey (Alt+L). In order to edit the data in a feature choose the Feature Variables button from the Modify Menu. You will be asked the hole feature that you want to edit. After picking it you will get several choices as seen in Figure 3.

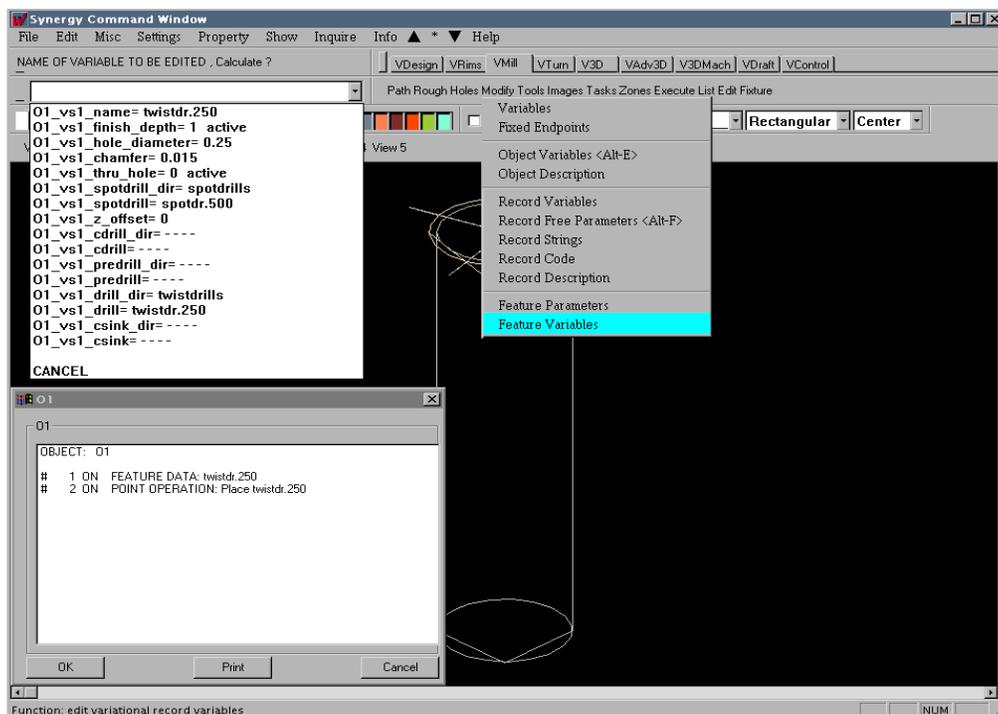


Figure 1.3 Editing feature variables. The menu choice, variables to edit, and a list of the records.

## Variable names

The following are descriptions of variables available through the hole features. Try modifying them on your own in order to get different hole feature output. Also, read in one hole feature twice. See how modifying one hole feature does not affect the data of the other hole feature.

**Name** – This is the name of the hole feature. You can change this to supply a unique identifier to the hole feature. In some instances it may be necessary to read a hole feature more than once. Therefore, a unique identifier will help you differentiate between a .500 drilled hole at level 0 and a .500 drilled hole at level –1.

**Finish\_depth** – This is the depth of the hole. Since it specifies a depth and not a z location, its value is assumed to be negative. Therefore a depth of 1 will place a hole that is –1 in Z. This value determines the bottom of the hole. A drilled hole’s finish depth is dependent upon the value of “Thru\_hole”. If the hole is a thru hole, not blind, then the tool tip finish depth specifies the depth to the top of the tool point. The tool point is then added to the finish depth to get the overall depth of the drill.

**Hole\_diameter** – This determines the size of the hole. It is standard and changing it will have no effect on the holes you see on the screen.

**Chamfer** – This will specify the size of the chamfer to be placed around the top of the hole. This is applied, in conjunction with the hole diameter, to either countersinks or spotdrills to determine their depth.

**Thru\_hole**—This can be one of two values. It is a true/false switch with 1 (true) being a thru hole and 0 (false) being a blind hole. A thru hole will automatically add the distance of the tip of a pointed drill onto the value “Finish\_depth”.

**Z\_offset**—This value specifies that the top level of the hole is somewhere other than where the hole was placed. If you have flat geometry, you may pick a circle for placement. However, the actual location of that hole may be 1 inch below the top of the part. So you would specify this value to be –1.

If you create custom hole features, you may find it necessary to include free parameters. The option to modify this is found under the Modify menu right above the Edit Feature Variables button.

Also included in the variable list are choices for different tool combinations. There is a possibility of using 5 different tools during a drilling operation. Mostly you may use anywhere from 1 to 4 of these tools. Using these variables at run time is not the recommended use of the system. It is recommended that you create a library of hole features for use at job creation. That will be covered later. For now, there are 2 sets of variables for each type of tool. One specifies the directory in the tool library where that type of tool can be found and the second is the tool file as found in that directory of the tool library.

**Spotdrill\_dir , Spotdrill** – Specifies the spotdrill to be used for starting a hole and chamfering.

**Cdrill\_dir, Cdrill** – Specify a center drill to use to create a start hole. (Generally, use this or a spot, not both)

**Predrill\_dir, Predrill** – For larger holes a preliminary drill will drill to depth to create a starting hole.

**Drill\_dir, Drill** – The drill to be used to create the hole.

**Csink\_dir,Csink** – A countersink to be used to place a chamfer on a hole.

The hole features are designed for a wide range of tooling possibilities. So, in almost all cases, some of the tooling available in a hole feature will not be used. These unused tools are then “blanked out”. The “blanked out” tools will have a value of either “- - -” or “none”. That way that tool will not be used for that set of holes. Generally the tool combinations for drilling may be a combination of either;

a spotdrill **OR** a centerdrill with a countersink *and*, a drill with the possibility of a pre drilled hole

or,

a single drill for carbide insert drills that self-start a hole.

There are 5 commonly used hole features: drills, taps, dowels, capscrews, leaderpins. Most hole features the are customized by users are similar to these. The similarities are shape, tooling, and manufacturing practice. Although these differ from user to user, the 5 common applications are ample for most people to modify and create their own hole features. To get a sense of how they work, the unique variables for each are described below (common variables have been left out):

## Taps

**Tap\_depth** – The specific depth of the tap. The depth of the drill is not affected by this number.

**Tap\_dir, Tap** – Specifies the location of the tapping tool.

## Dowels

**Ream\_depth** – Like the tap depth it specifies the depth of the reamed part of the hole.

**Reamer\_dir, Reamer** – Specifies the reamer. This tool will go to ream depth.

**Plungemill\_dir, Plungemill** -- Specifies that an endmill should be used to ream depth.

## Capscrews

**Finish\_allowance** – For a roughing endmill. This specifies the amount of stock on the head diameter.

**Bottom\_allowance** – For a roughing endmill. This specifies the amount of stock on the bottom of the head.

**Socket\_head\_dia** – Specifies the c'bore diameter for the head. (Not applicable if a counterbore is used on the head)

**Socket\_head\_depth** – Specifies the depth of the counterbore for the head.

**Second\_chamfer** – Not used. Specifies the chamfer at the head/pin corner.

**Cbore\_dir, Cbore** – Specifies if a counterbore should be used to create the socket head.

**Plungemill\_dir, Plungemill** -- Plunge an endmill to the counterbore\_depth.

**Roughmill\_dir, Roughmill** -- Round cut endmill. Roughs the counterbore

**Finishmill\_dir, Finishmill** -- Round cut endmill. Finishes the counterbore.

## Leaderpins

**Counterbore\_diameter** – The diameter to mill for the head of the leaderpin

**Counterbore\_depth** – The depth to mill for the head of the leaderpin

**Cbore\_rgh\_allow** -- The finish allowance for roughing the counterbore.

**Counterbore\_chamfer** – The chamfer around the top of the counterbore. Applies to csinks or chamfermills.

**Boringbar\_dir, Boringbar** – A boringbar to be used in the thru hole for the leaderpin.

**Chamfermill\_dir, Chamfermill** – The chamfering endmill around the top of the counterbore

**Chamf\_add\_depth** – Additional depth added to the calculated depth of the chamfermill. Gives more bottom clearance.

**Plungemill\_dir, Plungemill** – Plunge an endmill to chamfer the head/pin corner.

You will note that every tool has different sets of tool directory and tool files available for editing. In this way you can change the tools associated with a hole feature. However, always having to change the tools is a hassle. It is not recommended that you do this while trying to create a job. That's where Knowledge Based Manufacturing starts. You try to capture the knowledge of you machining practices and keep them locked. That way you can use them over and over without having to edit the hole feature every time you create a job. You capture your knowledge by creating your own hole features.

## SECTION 2. Creating Custom Hole Features

### How to create a hole feature

Creating a custom library is one of the major advantages to owning Synergy. A hole feature library allows you to use the knowledge gained over years of manufacturing. This is useful for easily creating jobs, tools sheets and process plans. Also, when training someone new, it is easy to show them how your company machines its parts. You can create a custom hole feature library in one of two ways. First, you can use the standard Synergy hole features with your own data. Second, you can modify the hole feature strategy to create a combination of custom data and custom code.

### Modifying feature data

There is a database of information held for each type of hole feature. These entries can be modified or deleted and new ones can be added. The data is contained in information packets that are stored in the directory;

*/usr/weber/database/properties/HoleFeatures.d*

under this directory there are hole features for metric and English units. Under that are the sub-directories for each type of hole feature. It is important to note that there are separate hole features for English and metric just like the tool library. The information in the packets is used to set the default values for the variables described in Section 1. Standard values such as tool file are specified but variables such as Finish\_depth are only defaulted to an answer and are overridden by the user. You can modify the packets directly or by using data tables. The following text will take you through the process of modifying a feature database using data tables.

First, start a new Inch job in Synergy. Choose to import a table as in Figure 2.1.

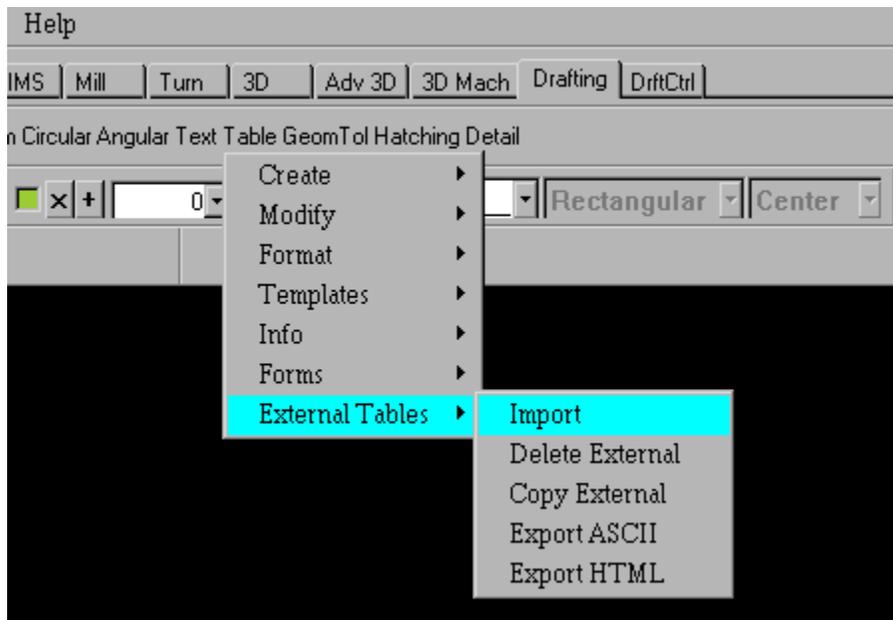


Figure 2.1 Importing a table. Design->Drafting->Table->External Tables->Import.

This will pop up a file box. The root of the tables directory that you are now in is:  
*/usr/weber/database/tables*

Under there go into the HoleFeatures.d directory and choose Inches (for purposes of demonstration choose Inches) or Metric. The subdirectories contain information for each type of hole feature. For the purposes of this example choose the drilling subdirectory.

You will see the choices for two tables. Each feature type has these two tables. The first is marked drilldata (tapdata etc.) and is no longer used by the system. The second is drillvalue. Choose drillvalue and use free point mode to place the table on the screen. Now import a property table by using **Design->DrftCtrl->Packets->Create->Make Property Table** and place it under the drillvalues table. This creates a tabular representation of the hole feature properties. It should look something like Figure 2.2 (except for the two colored rectangles).

Figure 2.2 Table Layout

Figure 2.3a and 2.3b are what is the yellow area above. Figure 2.3a shows the drillvalues table and Figure 2.3b shows the top of the property table. Figure 2.3c is a close-up of the Red outlined area and represents an area where the columns start changing value. This is a section of the property table you will concentrate on changing. This is how you will see the different ways a single hole feature can be used to output two distinct types of hole features.

Table path:	HoleFeatures.d/Inches/drilling/drilldata
	HoleFeatures.d/Inches/drilling/drillvalues
Directory:	HoleFeatures.d/Inches.d/drilling
Overwrite:	Yes
Class name:	Inch Drilled Holes
Class id number:	90001

Figure 2.3a The drillvalues table.

Name	Spotdrill_dir	Spotdrill	Drill_dir	Drill	Predrill_dir	Predrill	Drill_dir	Drill	Csink_dir	Csink	Hole.diameter	Finish_depth	Chamfer	Z.offset	Thru.hole
twistdr.0760	spotdrills	spotdr.500	----	----	----	----	twistdrills	twistdr.0760	----	----	0.076	1	0.015	0	0
twistdr.0781	spotdrills	spotdr.500	----	----	----	----	twistdrills	twistdr.0781	----	----	0.0781	1	0.015	0	0

Figure 2.3b Top of the property table. This shows the entries that are contained in the table. The entries correlate to the variables used in the hole feature strategy.

twistdr.5781	spotdrills	spotdr.750	----	----	----	----	twistdrills	twistdr.5781	----	----	0.5781	1	0.03	0	0
twistdr.5938	spotdrills	spotdr.750	----	----	----	----	twistdrills	twistdr.5938	----	----	0.5938	1	0.03	0	0
twistdr.625	spotdrills	spotdr.750	----	----	----	----	twistdrills	twistdr.625	----	----	0.625	1	0.03	0	0
twistdr.6562	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.6562	countersinks	csink_1.0	0.6562	1	0.03	0	0
twistdr.6875	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.6875	countersinks	csink_1.0	0.6875	1	0.03	0	0
twistdr.7188	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.7188	countersinks	csink_1.0	0.7188	1	0.03	0	0
twistdr.750	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.750	countersinks	csink_1.0	0.75	1	0.03	0	0
twistdr.7656	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.7656	countersinks	csink_1.0	0.7656	1	0.03	0	0
twistdr.7812	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.7812	countersinks	csink_1.0	0.7812	1	0.03	0	0
twistdr.8125	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.8125	countersinks	csink_1.0	0.8125	1	0.03	0	0
twistdr.875	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.875	countersinks	csink_1.0	0.875	1	0.03	0	0
twistdr.9062	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.9062	countersinks	csink_1.500	0.9062	1	0.03	0	0
twistdr.9219	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.9219	countersinks	csink_1.500	0.9219	1	0.03	0	0
twistdr.9375	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.9375	countersinks	csink_1.500	0.9375	1	0.03	0	0
twistdr.9844	----	----	centerdrills	N0drill	----	----	twistdrills	twistdr.9844	countersinks	csink_1.500	0.9844	1	0.03	0	0
twistdr1.00	----	----	centerdrills	N0drill	twistdrills	twistdr.625	twistdrills	twistdr1.00	countersinks	csink_1.500	1	1	0.06	0	0
twistdr1.0312	----	----	centerdrills	N0drill	twistdrills	twistdr.625	twistdrills	twistdr1.0312	countersinks	csink_1.500	1.0312	1	0.06	0	0

Figure 2.3c Section of the property table.

## Hole feature table -- values

The drill values table contains information that describes the data being stored. Since you can answer questions in the system by choosing text on the screen, this table is a quick and easy way to answer the questions involved in creating hole features. The first two lines describe the file path of the two tables that you placed on the screen. Since you are going to modify the tables to create your own set of data change the *drill* part of the file names to be whatever you want. A sample might be to change it to be *your company's initials* and *drill* so that Weber Systems would get *wsdrillvalues*. This will create unique filenames to store down the tables later. To modify a table entry, go to **Drafting->Table->Modify->Edit Entry**. Pick the boxed in area of the text ( cell ) and enter the new data. The old information should come up in the input line so that you can edit the entry.

The second part of the *drillvalues* table contains information that describes the data packets. These will be used later when we transfer the table data into hole feature information packets. Leave this section alone at this time.

## Hole feature table -- properties

Figure 2.3b showed the top of the property table. The headings of the table columns correspond directly with the variables that you saw when you edited a hole feature back in section 1. So we will try to refer to the headings of these columns as variables since that is what they represent. Each row in the table is a different combination of the variables. Mostly the variables that change from column to column are the tool values as seen in Figure 2.3c. In this way you can create many unique hole features. The first column is the name column. This represents the name of the hole feature and also the file name of the hole feature information packet. **This name has two restrictions, it cannot have spaces and it cannot have minus signs (“-“).**

Zoom in on the row for the twistdr.125. You will see it has a half-inch spotdrill. The two spotdrill variables represent the directory and the filename of the spotdrill. Although the directory name seems unnecessary, it is used when you have a custom library that does not follow the Synergy Tool Library layout.

This provides another level of user flexibility. In the centerdrill section (cdrill variable) there is no definition. There cannot be a blank entry in the table so “- - -” is used to show no value. The same goes for the predrill and the countersink (csink). As explained in section 1, the entry “- - -” will not disallow that tool from being used when the twistdr.125 is chosen. The other tool that is defined is the drill, which is usually (I won't say always) necessary since that is the purpose of the hole feature. The other tool entries are support tools that create start holes and chamfers. That is all dependent upon your specific needs. After the tooling is variable fields for hole\_diameter, finish\_depth, chamfer, z\_offset and, thru\_hole. Since finish\_depth and thru\_hole are set by questions, the values specified here are only defaults. For example, if you drill to 1.25 inches 80% of the time then you would set finish\_depth to be 1.25 and that would show up in the “REP=” field of the question when you place the hole feature. Also, if all your holes are always blind then set thru\_hole to be 0.

## **Table editing routines**

The reason this table exists is so that you can create your own set of hole features. Earlier, you changed the names of the tables so that unique tables could be stored down and saved. The default table can be modified to fit your needs. Under *Drafting->Table->Modify* there are several useful options.

**Delete Row --** Removes any row that you don't need in your table.

**Add Rows --** Appends empty rows at the end of the table so you can enter your own values.

**Insert Rows --** Inserts empty rows where you specify in the table so you can enter your own values.

**Edit Entry –** Modify a cell of the table

**Copy Row –** Copy an entire row of data into some other row. Useful for bypassing manual entry.

**Copy Entries –** Copy cells from one part of the table into another. Useful for bypassing manual entry. This uses a stretch box to pick a box of entries at once. To use it, pick the upper left corner box you wish to start with and drag to the lower right box where you want to end. That chooses multiple cells to copy at once. To choose one cell to copy click twice in the cell to choose only that cell.

## **Editing table entries**

You notice that there are three different types of drilled hole features. The first has spotdrills and twist drills. The second has a centerdrill, twistdrill and, a countersink. The third has is the same but with a pre-drill. Examples of manipulating the table will be laid out here. But, first choose *File->Save Options->Backup*. This will save the file so that you can come back to it to make your own changes. For now you will manipulate it to show some different editing techniques.

Whenever referring to a hole feature, it is proper to refer to its name in the first column of the table. You will convert the twistdr.6562 thru the twistdr.7188 so they use spotdrills only. All table edit functions can be found under the *Design->Drafting->Modify* menu.

- 1) Choose Copy entries.
- 2) Pick the table.
- 3) Pick the twistdr.5781 spotdrills entry (2<sup>nd</sup> column).
- 4) Drag the box down two more rows and over one column to the right. This will be the spotdr.750 box in the twistdr.625 row.
- 5) Pick that box.
- 6) Choose the twistdr.6562 2<sup>nd</sup> column. This places the copied entries in the table.
- 7) Hit the undo button.
- 8) Do steps 2-6 again. This time drag the box over 3 columns to the “- - -” in the centerdrill column. (This will enter the spotdrills and blank out the centerdrills)
- 9) Do the same steps but now do them to the countersinks column.

Now that your targets are done, the new spotdrills are too large for the holes you want. So you will change those three entries to read spotdr.875 instead of spotdr.750.

- 1) Choose Edit Entry.
- 2) Pick the spotdr.750 cell of the twistdr.6562.
- 3) The old entry will come up in the input line. You can choose to re-type it or click in the window and replace the 750 and with .875. (Typing will overwrite a highlighted entry).
- 4) Now that the entry has changed. Choose the next cell.
- 5) In order to answer this question, pick the previous cell that was changed in step #3. Synergy can use the picking of text to answer questions. Instead of typing the same value in again, pick it from existing data in the table.
- 6) Repeat #4 and #5 with the next cell.

Let's take the example where you do not need any drills over .875 in diameter. You will delete these tools from our small section.

- 1) Choose Delete Row.
- 2) Pick any cell in the twistdr.9062 row. (The row will highlight)
- 3) The question now asks to Undo (you didn't mean to delete it) or Proceed (go ahead and delete). Choose Proceed.
- 4) Now that the row is gone. Pick the next 5 rows one at a time.
- 5) Choose Proceed. This will delete multiple rows at one time.

Hole features can be used in different instances. If you look at the twistdr.750 row, you will notice there is a countersink available. There may be an instance when you want to remove the countersink. This can be accomplished in two ways. First, when the twistdr.750 hole feature is used, edit the csink variable to be "none". This will eliminate the tool. The other way is to create two hole features. One with a countersink and one without a countersink.

- 1) Choose Insert Rows. This will insert 1 or more blank lines into the table. For rows at the end of the table use Add Rows).
- 2) Choose the twistdr.7656. This is the row you want to insert the blank rows before.
- 3) Enter 1 row to insert.
- 4) Choose Copy Row.
- 5) Pick the twistdr.750 cell. This is the row you want to copy.
- 6) Pick the first blank cell in the new row. This is the destination and you will get a copy. You will have two twistdr.750 rows. You will change the second one (the copy).
- 7) Choose Edit Entry.
- 8) Choose the twistdr.750 cell in the copied row.
- 9) Add a "\_nocs" to the end of the twistdr.750 so it becomes twistdr.750\_nocs.
- 10) Choose the countersink directory and countersink tool entries in that column and replace them with "- - -" from another cell.

This method can be used to create different entries for

- Different tool combinations
- Different depths, so a standard hole feature will always go to a specific depth
- Blind and thru holes

You can save this file as reference for later. At this point you can create you own hole features or continue on with this current exercise. If you want to create you own hole features go to **File->Open->Recover Backup** to restore the tables as they looked previously. You can now modify the table to contain the information that you need to create your own hole feature library. When you are done, return to this point and continue on.

## Modifying the drillvalue table

The table now contains the information that describes all our different scenarios for drilling. We now want to create the info packets that will drive our hole features. Zoom in on the drillvalues table (small table at the top) and a slight portion of the property table. We need to edit the table so that we can get a unique set of hole features. The entries in the drillvalues table are in two sections. The first, covered earlier, is the names of the table files. The second section is the data used to answer questions for storing hole features.

**Directory** – The directory under `/usr/weber/database/properties` where the hole features are to be stored.

**Overwrite** – Used only if the directory already exists. Asks if you want to overwrite entries that already exist or not. Also, the Delete option is available. This is the preferred answer to the question if you want to overwrite an entire directory. Since you can create and re-create these hole features you may have some that you no longer want to use. This can be accomplished by deleting entries in the table. However, when you go to store down the packets, old files that are no longer in the table will still exist in the packet directory. Using delete will wipe out the packet directory first before storing. If you wish to use the Delete option change the entry in the drillvalue table from Yes to Delete.

**Class name** – Packet information describing the type of hole feature.

**Class id number** – A unique packet identifier that further classifies the hole feature. You may re-use these ID numbers for the standard Synergy hole features. If you create your own hole features create a unique ID number. A list of the ID numbers can be found by using *Design->DrftCtrl->Packets->List->List External Class ID Numbers*.

Besides changing the overwrite switch, you also want to change the directory entry. Since Synergy always updates the **HoleFeatures.d/Inches.d/drilling** directory, you will want to create your own directory. So, Edit Entry of the directory path and change the HoleFeatures.d part of it and add your company's initials to the beginning (like with the tables). Now we will have a unique storage directory that will not get overwritten by Synergy. Now you are ready to store the hole feature data.

## Creating a custom hole feature directory

Go to the *Design->DrftCtrl->Packets->Create->Make Properties From Table*. The box that comes up on the screen is asking for the directory where you want to store your hole feature. However, we will bypass this by choosing the cell that specifies the hole features packet directory. Move the box that just popped up down to the lower right hand corner so it's out of the way. Look at the drillvalues table. Each question that comes up will be answered in succession by the table. So to answer the PACKET CLASS DIRECTORY question pick the table cell that contains HoleFeatures.d/Inches.d/drilling (your changed entry). The first time that you run this script the Overwrite question will not be asked. So, use caution when answering. You can use the overwrite field for successive saves of the hole features. Keep answering the questions down the list. When Synergy asks for what table to use, choose somewhere inside the large table (property table). You have now stored the hole features. You can repeat this step by walking through it manually to see where the hole feature directory is and, to see the Overwrite question.

When making other hole features, change the name of that directory so that the same HoleFeatures.d directory path is used throughout. This allows all the hole features to come from the same directory tree. However, you must specify you new directory so that the system will use it. You will do this by adding the following line to your userconstants file (if you do not have a custom one create it);

```
V1=CreateVariable("hole features sub dir",WsSystemVars,"HoleFeatures.d")
```

Change HoleFeatures.d to the name of your new directory. Also the following line can be added to any CAD settings file;

```
SystemVars.hole features sub dir:   HoleFeatures.d
```

## **Storing the table information**

We now need to store the table information in order to save it. It needs to be noted again that this is just a convenience function for quickly editing large groups of data. The actual data that gets used by the hole feature is stored in the info packets. Earlier, you changed the table path names. Now choose the *Design->Drafting->Table->External Tables->Export ASCII*. For the file name choose the drillvalue path and pick the drillvalue table.

## **Recreating hole features**

The table technique can be used over and over when creating hole features. It is not the only way. You can manually edit a hole feature file by using *Design->DrftCtrl->Packets->Create->Create Property*. This will show a list of all the data in an individual hole feature packet. Changing the name field will create a new packet upon storage. This allows you to create a single packet when needed without having to use the tables. Also, you can just use the function to modify existing data for one packet. Such as the default finish\_depth.

## Section 3. Creating custom hole feature scripts

### Hole Feature Script Basics

Creating a custom hole feature file can be done in two ways. The most direct of which involves editing the file directly. The second is to create the hole feature thru the system. The best way is to edit the file directly. It requires a knowledge of VDM techniques and Synergy script code. However, we will try to cover those basics in this section. Go to *Variational->Vcontrol->Strategy->Edit->File*. The hole feature file directory is located in; `/usr/weber/database/strategies/`

- 1) Choose **holefeatures**.
- 2) Under that there are two directories; **inches** and **metric**. One of these directories is chosen based upon what unit of measure you use to program your job. For our purposes go to inches.
- 3) Choose **drilling**.
- 4) There are two hole feature scripts here. Choose **drillmaster**. The pdrillmaster script is for mill-turn power tooling. (Note: Older systems may have individual file names that start with twistdr. They can be removed since they are outdated)

### The hole feature script: drillmaster

The hole feature script is broken down into sections. Each one will be explained as you read along. What you will see is a Synergy strategy. This is the Variational Design Module (VDM) technique of teach-in and play-back. It will look very different but, for all differences there is a reason. The strategy has a structure that must be followed in order for it to work. Following this structure is easy when you know what to do and why things are a certain way. That's what this document will accomplish.

This script has its origins in Synergy. It was created by teaching-in geometry through the graphical system. However, over the years it has seemed easier to manipulate the script through editing rather than trying to create it through the system. The next few pages will go through the script by showing a section first and then explaining it afterwards. The sections are self-contained any information outside of them is ignored.

```
START HEADER:  
Strategy file name: /usr/weber/database/strategies/holefeatures/inches/drilling/drillmaster  
Instruction 1:  
Instruction 2:  
END HEADER:
```

This is the header information associated with the file. It contains the name of the file and two lines for instructions. The instructions are not commonly used with holefeatures. The name should be changed when you are creating a new file. A new file can be used through the *Vdesign->Holes->User defined* button.

```
START STRATEGY SETTINGS:  
info packet strategy: 1  
edit strategy vars: 2  
END STRATEGY SETTINGS:
```

The strategy settings are values to get the strategy to behave in different ways. The different values that can be used with hole features are:

- **info packet strategy:** This is a switch that is 0 for false and 1 for true. It describes that the hole feature will use info packets to answer most of the variables. This value should not be changed.
- **edit strategy vars:** Three settings describe how the variables will be ask when the hole feature is used. 0 will not ask variables. 1 will pop-up a box with the variables. 2 will ask questions one at a time.

```
START ENTITY REFERENCES:  
END ENTITY REFERENCES:
```

Describes any entity that must be picked to implement the strategy. Not used in hole features.

```
START TOOL REFERENCES:  
END TOOL REFERENCES:
```

A different way to describe tooling non-variationally. Not used in hole features.

```
START PROPERTY DATA:
Hole feature dir: Inches.d/drilling
END PROPERTY DATA:
```

This describes the subdirectory where the hole feature packets are stored. This directory was specified in the drillvalues table earlier in the document. If you stored the packets under a different directory other than drilling, this value would have to be changed to match. The value in the table had **HoleFeatures.d** at the beginning of the path. That was for reference when storing down the packets. This value describes where the features are underneath the “Hole features sub dir” that you changed (or not) in userconstants. This way you can change the “Hole features sub dir” and not have to have a custom hole feature script.

```
START VARIABLE REFERENCES:
Xxx_ext_var_name: twistdr.3125: inactive: Name
Xxx_ext_var_spotdrill_dir: spotdrills: inactive: Spot drill directory
Xxx_ext_var_spotdrill: spotdr.500: inactive: Spot drill
Xxx_ext_var_cdrill_dir: none: inactive: Center drill directory
Xxx_ext_var_cdrill: none: inactive: Center drill
Xxx_ext_var_predrill_dir: none: inactive: Pre drill directory
Xxx_ext_var_predrill: none: inactive: Pre drill
Xxx_ext_var_drill_dir: twistdrills: inactive: Drill directory
Xxx_ext_var_drill: twistdrLq: inactive: Drill
Xxx_ext_var_csink_dir: none: inactive: Countersink directory
Xxx_ext_var_csink: none: inactive: Countersink
Xxx_ext_var_hole_diameter: 0.375: diameter inactive: Hole diameter
Xxx_ext_var_finish_depth: 1: depth active: Drill depth
Xxx_ext_var_chamfer: 0.015: inactive: Chamfer
Xxx_ext_var_z_offset: 0: inactive: Z offset value
Xxx_ext_var_thru_hole: 0: active: Hole type ( 0 )BLIND or ( 1 )THRU
END VARIABLE REFERENCES:
```

These are the variables used for this hole feature. You will notice a one-to-one correspondence with the headings in the property table. They are the same except for the Xxx\_ext\_var\_. That specifies that they are variables that are external to the strategy. In other words, the user can change them. If it is necessary to add your own variable at any time you can add it here. Just copy one of the other lines and change the name and default value. The default value is after the first colon. It is needed for strategies to be run but is not used in the hole features. The info packet that is chosen overrides this value.

There are 4 fields in each line of this section. Colons separate them. The first two have already been described. The third one is a field with multiple uses. If you look at the hole\_diameter line you will notice that there are two words, diameter and inactive. These are just 2 of the possible 9 values that can be in this section. They are;

- active** or **inactive** – One or the other is always specified. An active variable will have questions asked. An inactive variable will have no questions and must be changed through editing.
- global** – Variables are unique to each hole feature unless global is set. The cdrill variable in a tap is different than the cdrill variable in a dowel. The global switch use the same variable for both.
- numeric** – Specifies that a variable’s value is a number and not a string.
- string** – Specifies a variable’s value is a string and not a number. A string can be both letters and numbers (i.e. “123”). So, **string** is used to specify that this variable should be initialized as a string and not a number. This way a value of “ABCD” can replace “123”.

4 other names allow you to specify that a variable contains more than just numeric data. **diameter**, **radius**, **depth** and, **angle** are used to modify the variables so that circles, lines and, holes can be picked where appropriate.

```
START OBJECT: 01
```

This section contains the rest of the file sections. It goes from here to;

END OBJECT: O1

This is called the VDM object. It is the encapsulation of data for the hole feature. Objects contain records.

START RECORD:

Record descr: FEATURE DATA: Twistdrill Operation

This section is the beginning of the first record in the object. For hole features there is only one record. The record has a description specified at the beginning. This can be edited. However inside the code for the record there is a call to change it automatically to the name of the hole feature. This record goes until;

END RECORD:

That ends the first record. If another record were to be specified, it would follow this END RECORD with another START RECORD, then code, then another END RECORD. All of that would be contained in between the previously discussed START OBJECT and END OBJECT.

The rest of the script is the record of the hole feature. This same code can be seen when the hole feature gets used in the system by using *Vcontrol->Control->List->Record*. The breakdown will be on a line per line basis. The first section of code is generally not touched. The first main editing area is the Finish\_depth area.

```
Vstr=StringConstant(FindNamedVar("Xxx_ext_var_name",WsVariationVars))
```

The name variable is copied into the variable Vstr. StringConstant is a function used to create a temporary variable (Vstr) that is only active inside of this record. Whereas, all the variables discussed previously are always accessible. The Vstr is a Synergy script variable. Any of these temporary variables you will see are named V1,V2,Vstr etc. The common usage is to have a V as the first character. For this example Vstr will be equal to twistdr.250

```
Vname= StringConstant(JoinStrings("FEATURE DATA: ",Vstr))  
Stat= CurrentRecordDescr(Vname)
```

The Vname variable is created and then used to replace the record description with "FEATURE DATA: twistdr.250".

```
Vt1=StringConstant("none")  
IF UPC$(VariableString(Vt1))<> "NONE" THEN  
Vtinst=StringConstant(JoinStrings("---",Vt1))  
ELSE  
Vtinst=StringConstant("")  
END IF
```

A StringConstant can be changed just like a variable can be changed. Use the *Vmill->Modify->Record* Strings to edit the "none" in Vt1. This string is used to append onto tool names so that you get distinct tool names per hole feature. This is useful in AMO when you want to order when to use the twistdr.250 if it is used in more than one hole feature. This functionality is only used when Vt1 is something other than "none". The Vt1 portion will not show up when you edit Record Strings. All that shows up in the list is the word none.

```
V1=FreeParameter("Finish depth","Xxx_ext_var_finish_depth")  
V2=FreeParameter("Hole diameter","Xxx_ext_var_hole_diameter")  
V3=FreeParameter("Chamfer","Xxx_ext_var_chamfer")  
V4=FreeParameter("Thru hole","Xxx_ext_var_thru_hole")  
Mdef=HoleFeatureDef(WsDrillOperation,V1,V3,V2,V4)  
Stat=NameEntity(Mdef,"Xxx.Ent.Mdef1")  
Stat=SetMachDefDescr(Mdef,Vstr)
```

## Section 4. Hole Modification Functions

### Usage of hole features

Hole features are just a specific type of strategy. The only form they have to follow are the constraints of strategies and creating a hole group that can be placed. However that end result is achieved is only limited by knowledge of VDM. Below are some functions that can be used to create a hole to whatever size you need.

### Variables

You can expand your own hole features to include as many variables as necessary to accomplish your end result. You use the FreeParameters to add, subtract, etc. these variables together. Such as:

```
V3=FreeParameter("Cbore allow","Xxx_ext_var_rough + Xxx_ext_var_overcut")
```

This would add the variable Rough and Overcut together to make Cbore allow.

### Function Usage

There are many functions that can be used in constructing your own hole features. If you look at the drillmaster hole feature you will notice this section;

```
H1=DrillDefinition("Xxx.Ent.Mdef1",Tool)
Stat=ShareMachDefValue(Hpre,H1,WsHoleDefDepth)
V1=FreeParameter("Drill rap level","Xxx_ext_var_z_offset")
Stat=SetMachDefValue(H1,WsHoleDefZoffset,V1)
```

If there is a pre-drill and a drill in the same hole feature, this section comes into effect. The pre-drill gets created first, above this section. Then the drill gets created at the beginning of this section. After **H1** created, the drill shares its value for HoleDepth with the pre-drilled hole **Hpre**. Then the value for the Zoffset of the drilled hole is set to the value of **V1**. **V1** is set to the variable Z\_offset. The functions **ShareMachDefValue** and **SetMachDefValue** are two of the functions that are useful in manually constructing hole features. There are several pass parameters that can be used besides the ones shown above.

### Functions

The following are functions that can be used to create hole features.

```
Stat=SetMachDefValue(Hole, Feature to change, Value)
```

```
Stat=ShareMachDefValue(Destination Hole, Source Hole, Feature to share)
```

These two functions allow you to edit the Hole Definition (H1) directly. **SetMachDefValue** allows you to set a feature of a hole definition directly. **ShareMachDefValue** will share a feature of one hole and assign its value to another hole. The features of a Hole Definition are as follows:

**WsHoleDefDepth** – The depth of the hole.

**WsHoleDefDiameter**– The diameter of the hole.

**WsHoleDefZaddon**– An additional depth value. Can increase or decrease the depth of the hole.

**WsHoleDefZoffset**– The Z level of the top of the hole relative to the placement point.

**WsHoleDefAngle**– The tool tip angle of the hole.

```
Stat=ToolDiameter(Tool)
```

```
Stat=ToolPointAngle(Tool)
```

```
Stat=ToolCornerRadius(Tool)
```

```
Stat=ToolFluteLength(Tool)
```

```
Stat=ToolTaperAngle(Tool)
```

```
Stat=ToolTipLength(Tool)
```

These are functions that retrieve data from a tool. Those values can then be used in calculations.

**VariableValue(V1)**

This is used to extract the value of a FreeParameter.

**Stat= PutUserValue(*Tool,Field,Value*)****Stat= PutUserValue(*Tool,Field,Value*)**

This function is used to put fields back into the tool. The field value is one of the fields in the tool.

**GetUserValue(*Tool,Field*)****UserDataStr(*Tool,Field*)**

This function is used to get field values from the tool. The field value is one of the fields in the tool.