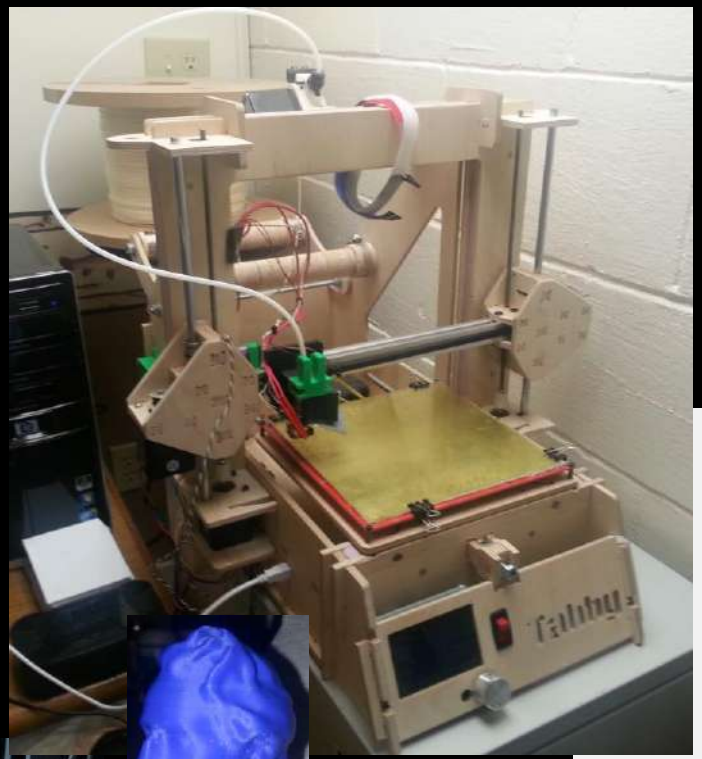
DO YOUR OWN THING WITH USER-DEFINED FEATURES (UDF'S)

Andy Meinert

MY HISTORY

- Born/ Raised in Peoria
- BSME 1989 U of Illinois
  - Engineering Co-op – NL Industries (2 semesters)
  - Project Engineer Intern – GE (2 years)
- Design/Test Engineer – Komatsu-Dresser (3 years)
- Pro/E consultant
  - PTC (7 years), Self-employed (2 years), NASCO (3 years)
- Design/Analysis Engineer – Leading Edge Engring (1 year)
- Black Belt/Project Leader – Caterpillar (12 years)

MY FUN



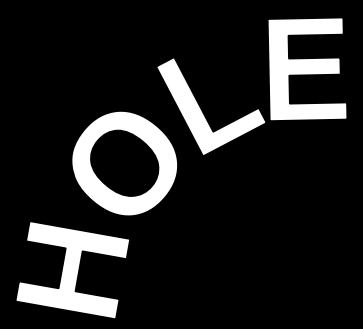
PURPOSE

1. To inspire Pro/E users to try a little-understood but powerful tool.
2. To help you get off to a good start doing so.
3. To send you home with something useful.
4. To amuse myself and get away from work for a while.

WHAT IS A FEATURE?

Feature:

- A discreet package of information whose purpose is to combine with others to build a model and which is only useful within the context of the model.
- The most essential level of knowledge that the model contains.
- The tools that generate geometry.



FEATURES CONTAIN

- mathematical algorithms
- references to other features
- Parameters (dimensions)
- definitions of geometry elements

PART NAME : BASE_FOAM
FEATURE NUMBER : 7
INTERNAL FEATURE ID : 5104
FEATURE NAME : MOTOR_CUTS

Parents

No.	Name	ID	Actions
2	BASE_Z	25	
4	BASE_Y	32	
5	Extrude 1	3824	

Feature Element Data - CUT: Extrude

No.	Element Name	Info
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Remove
4	Section	Defined

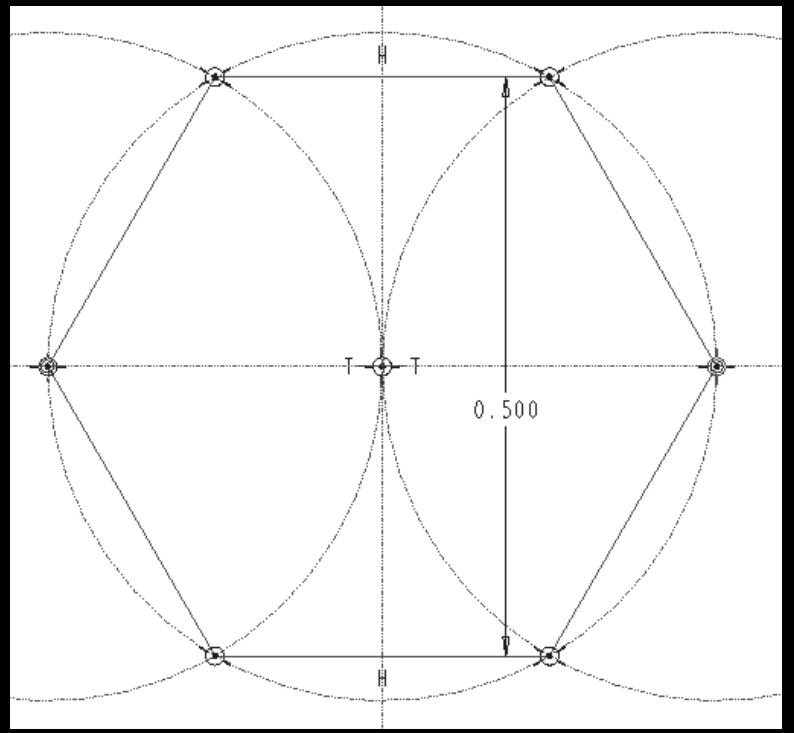
FEATURE'S DIMENSIONS:

Dimension ID	Dimension Value	Displayed Value
d1002	1.75 (0.1, -0.1)	1.8
sep_dist	23.8702 (0.1, -0.1)	23.9
d1011	13.894 (0.025, -0.025)	13.894

Relation Table

Relation	Parameter	New Value

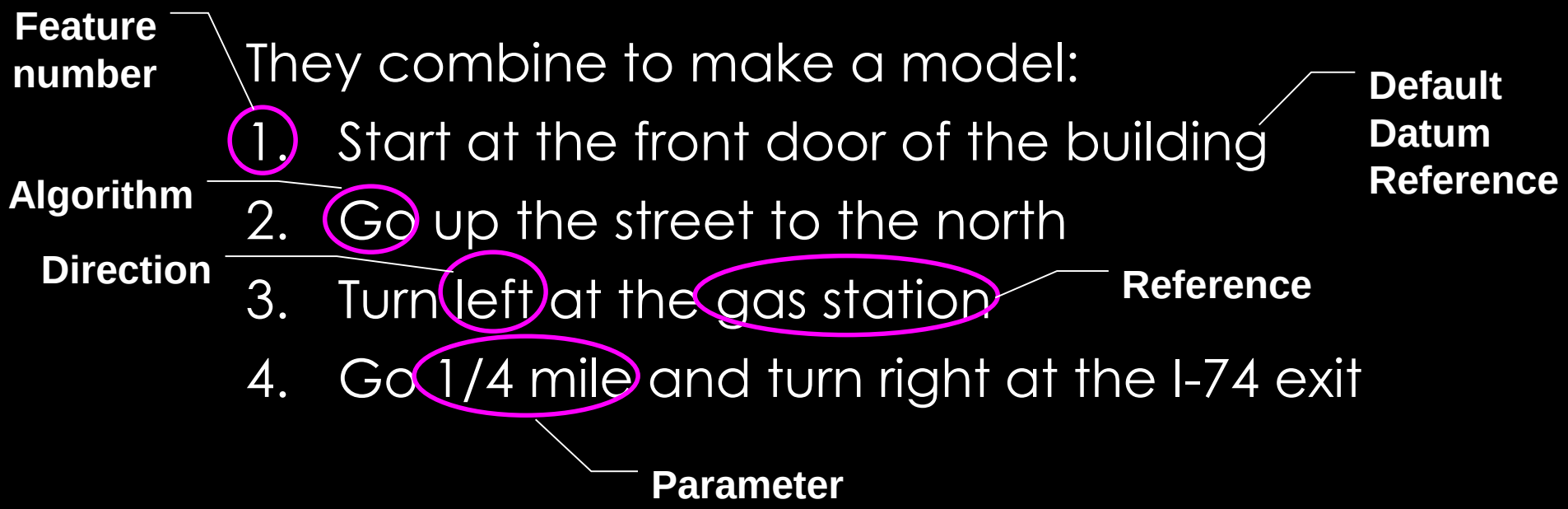
Part relations driven by this feature:



FEATURES: THE ANALOGY

Features are like the individual instructions you provide when telling someone how to get somewhere

- Have order, references, parameters, actions, directions

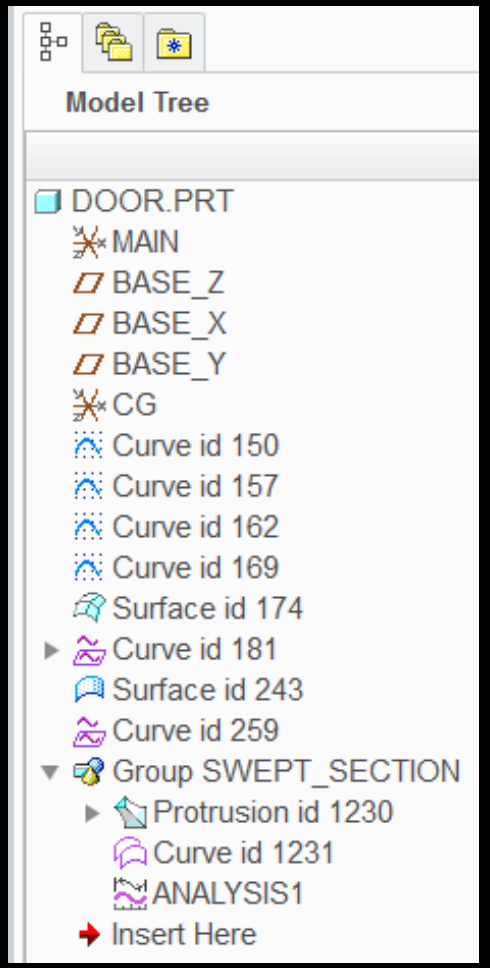


USER-DEFINED FEATURE

Definition: one or more features that a user has defined and then saved to the disk for reuse later

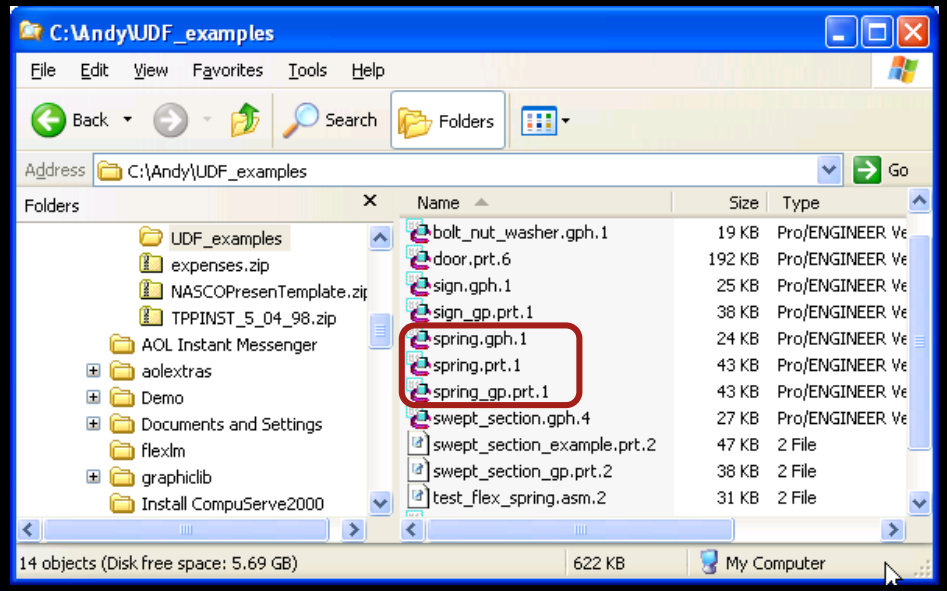
UDF's turn complex features into pick and place features

UDF's are considered "groups." They are groups of features that you find useful to reuse together.



UDF ANALOGY

Like in Windows editors you can copy a line of instructions to the clip board and then reuse it by pasting into a document later



Except the UDF is saved in a file. It doesn't go away when you shut down, etc.

BEHAVIOR

When you use ("Paste") a UDF in a model, the UDF acts based on its context:

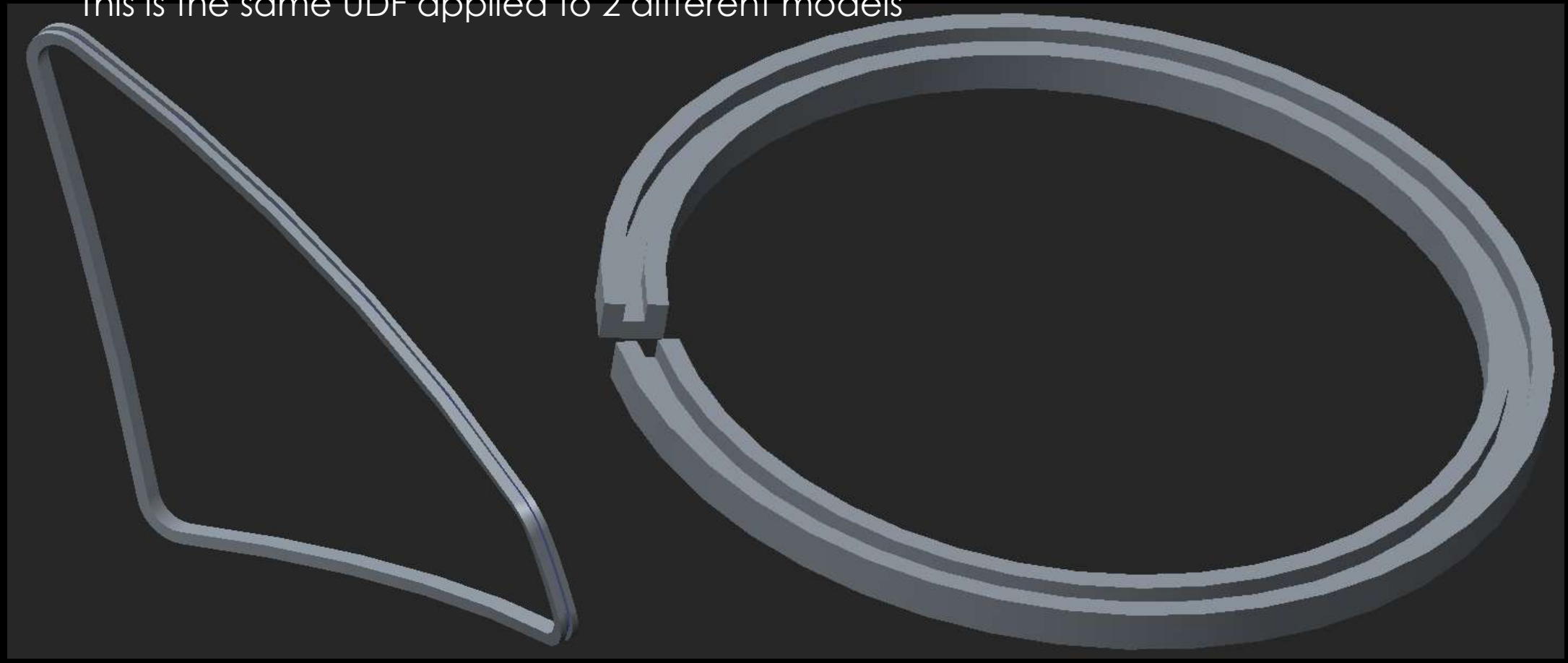
- References
- Parameters

You do not necessarily get the identical geometry as when you originally defined it, you get the same intent/knowledge.

The UDF can not be opened on its own. It is only meaningful within a model.

BEHAVIOR

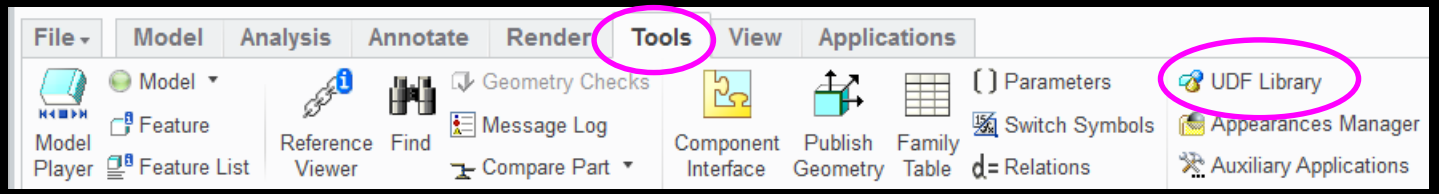
This is the same UDF applied to 2 different models

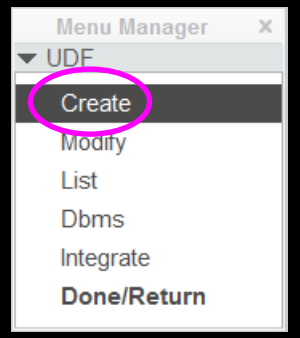


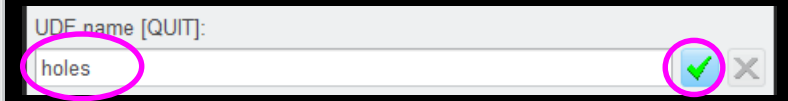
PROCESS

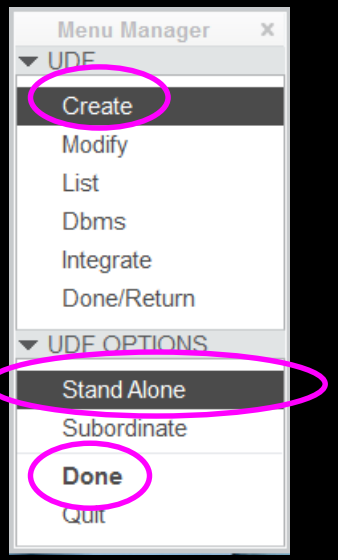
1. Identify the need
2. Design the UDF
3. Create a special part for defining the UDF
4. Create the features
5. Define the UDF (Tools, UDF Library, Create)
    a. Name
    b. Features to include
    c. Prompts
    d. Variable dimensions, parameters and elements
    e. Family Table
6. Use the UDF in models (Model, User-Defined Feature)

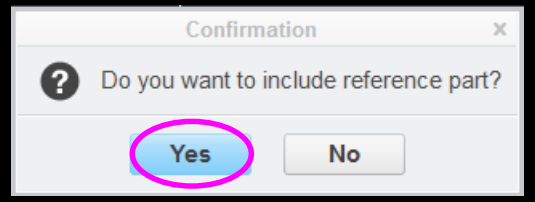
MENU PICKS – DEFINE THE UDF

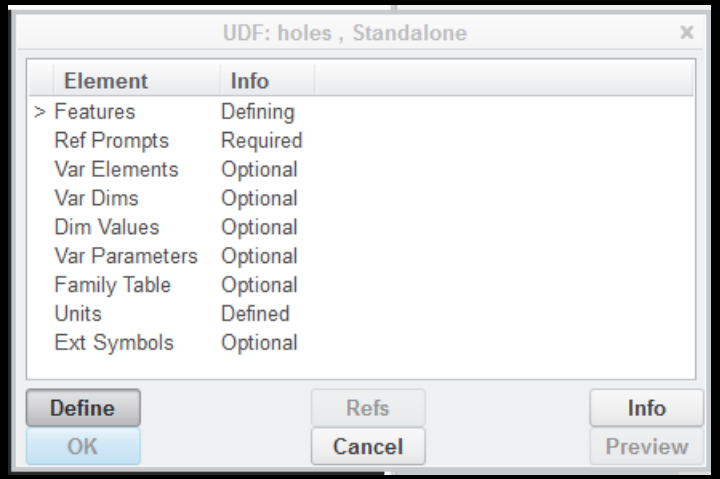




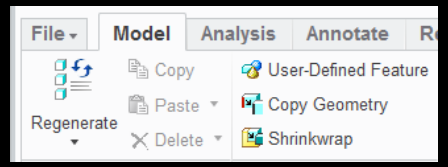


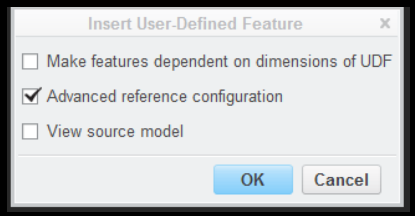


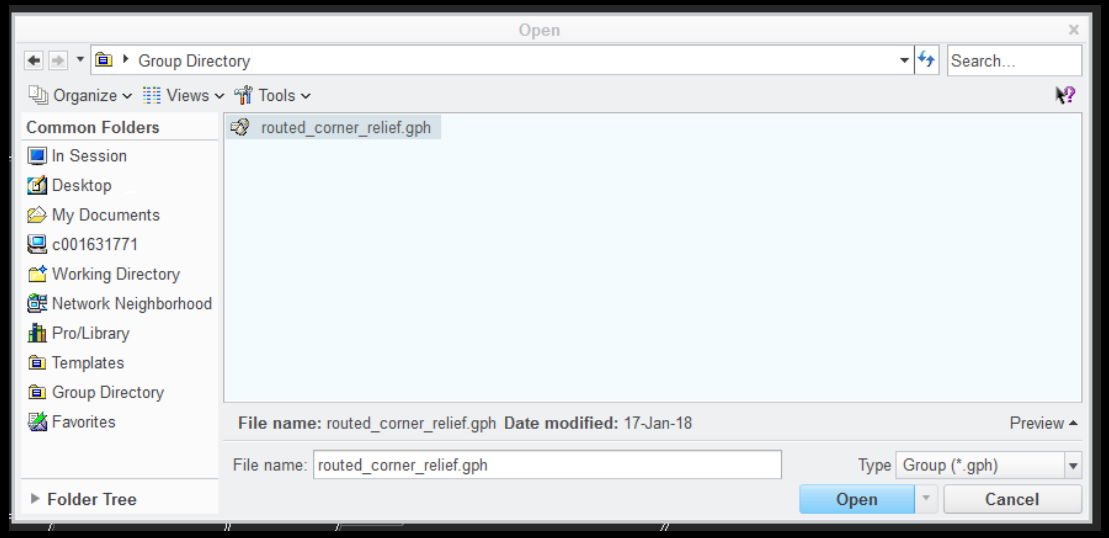


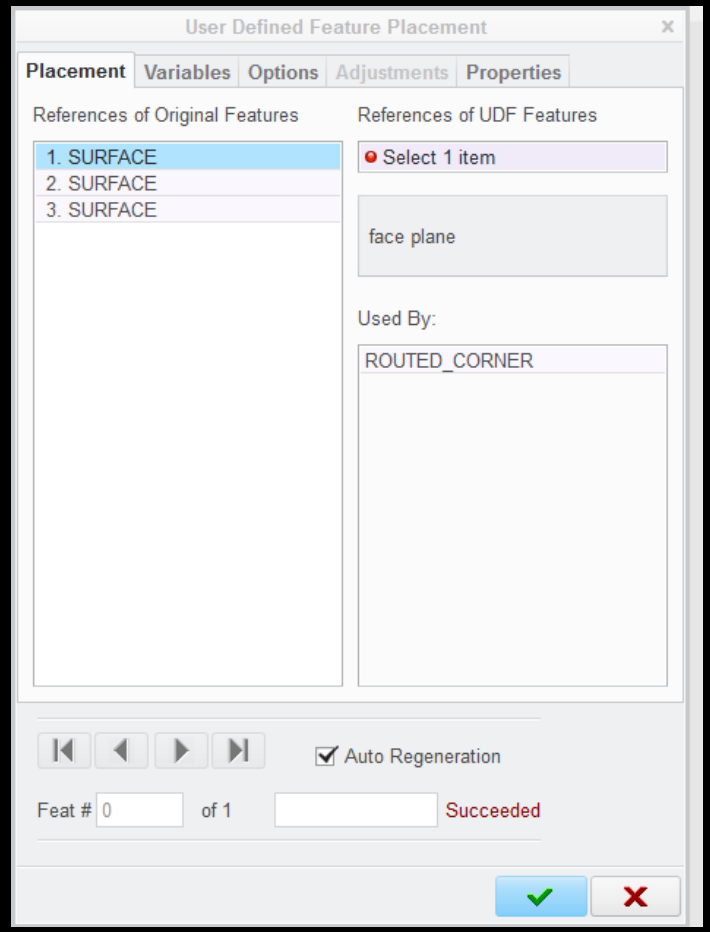


MENU PICKS - USE THE UDF









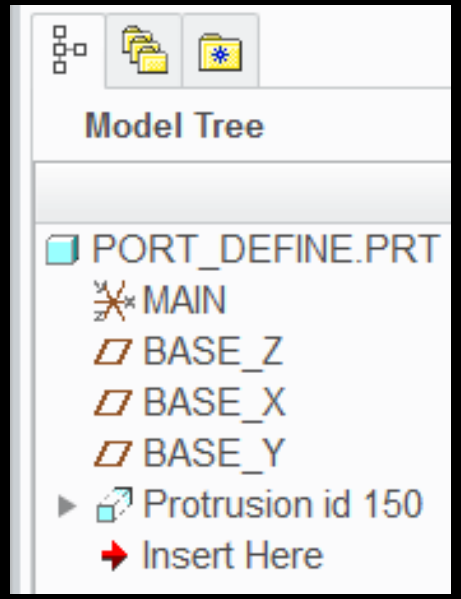
BEST PRACTICES

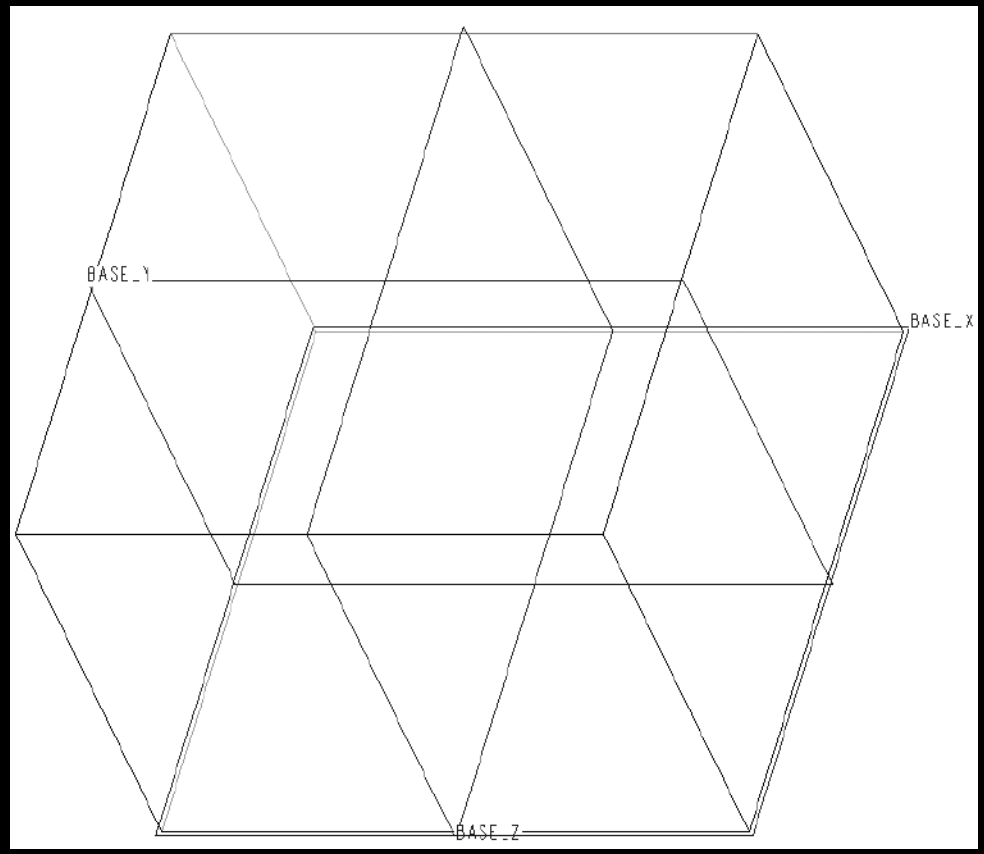
Create a separate part for defining the UDF rather than using a regular part.

- Single cube, cylinder, wall is best as a base
- Helps keep you from picking extra references
- Lets you define specific or generic features that you need
- Small easy model to save on the disk and reference later

BEST PRACTICES

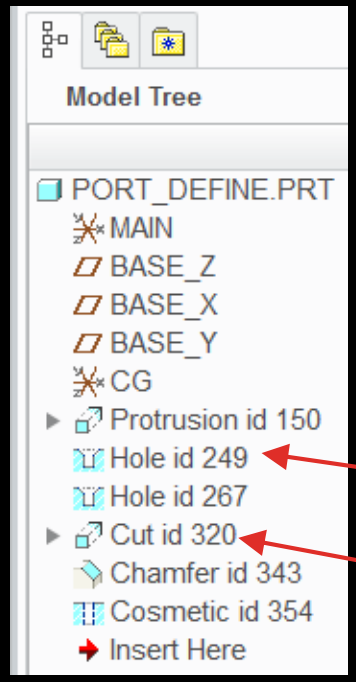
Only create the features you need for the UDF so you can avoid unnecessary relationships

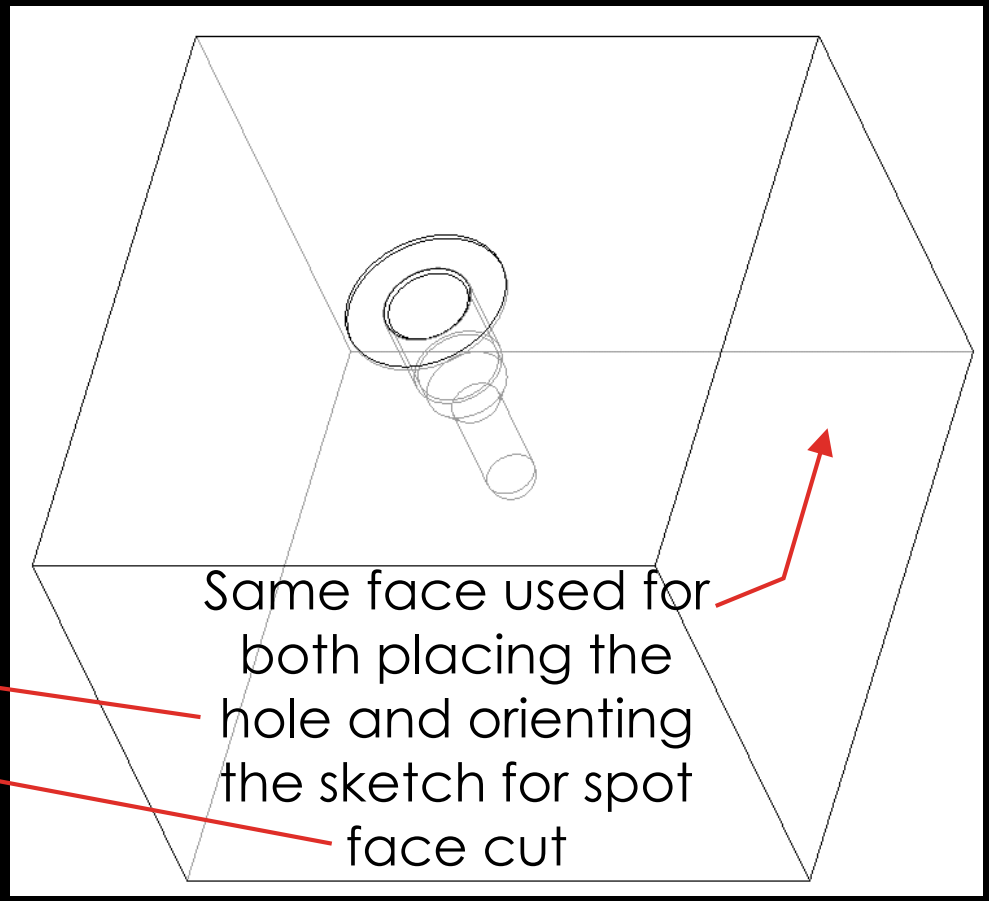




BEST PRACTICES

When appropriate, use the same references for each feature and tell Pro/E to only prompt for them once, not multiple times





BEST PRACTICES

Save the part with UDF when prompted

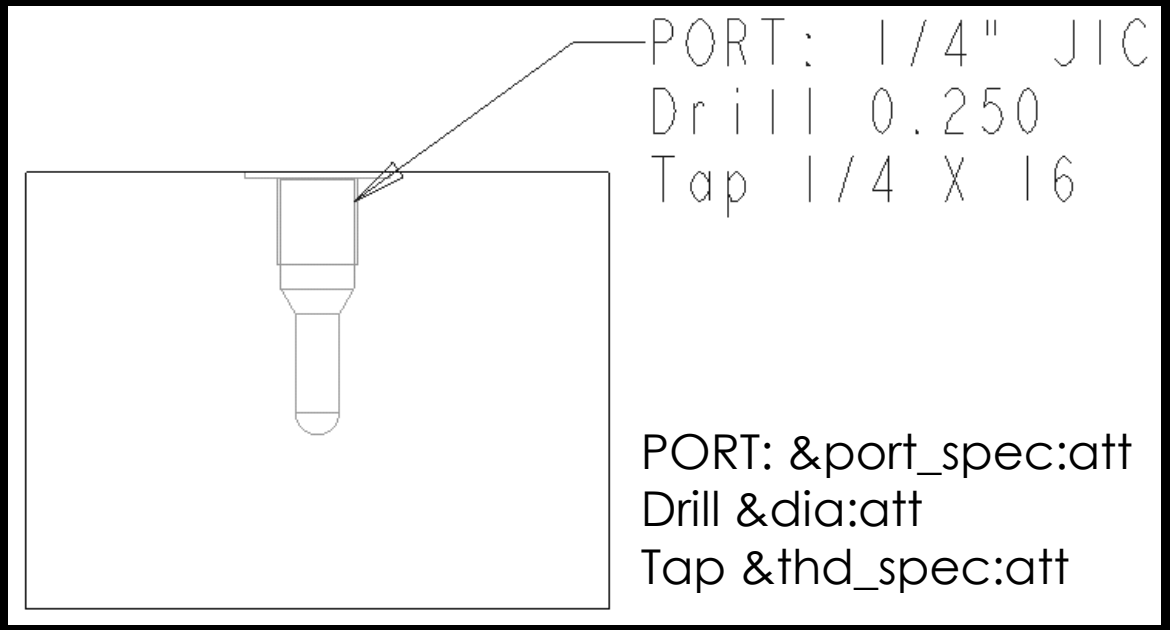
- allows for adding more features/capabilities to UDF
- Can show original refs when placing UDF

BEST PRACTICES

- Use descriptive prompts
  - “placement plane,” “horizontal reference plane,” “x-dim reference”
- Write feature relations, not part relations
- Make them Independent and stand-alone, not subordinate and UDF-driven unless absolutely necessary

TIPS

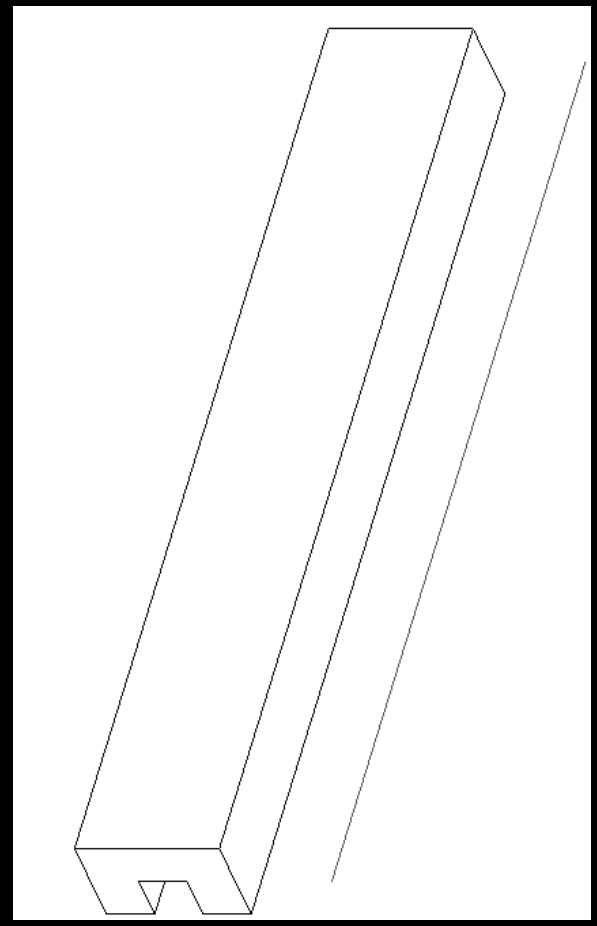
- Build/test/refine as much as needed
  - Expect it to be wrong the first few times
- Drive feature parameters from dimensions using feature relations then you can access info from them using saved notes, etc.



dia = d75

TIPS

- Build/test/refine as much as needed
  - Expect it to be wrong the first few times
- Drive feature parameters from dimensions using feature relations then you can access info from them using saved notes, etc.
- When sweeping along curves, use a single straight curve and “Loop” option for defining
    - Otherwise, you will be prompted for all the curves in the loop
- Analysis features add capability



TIPS

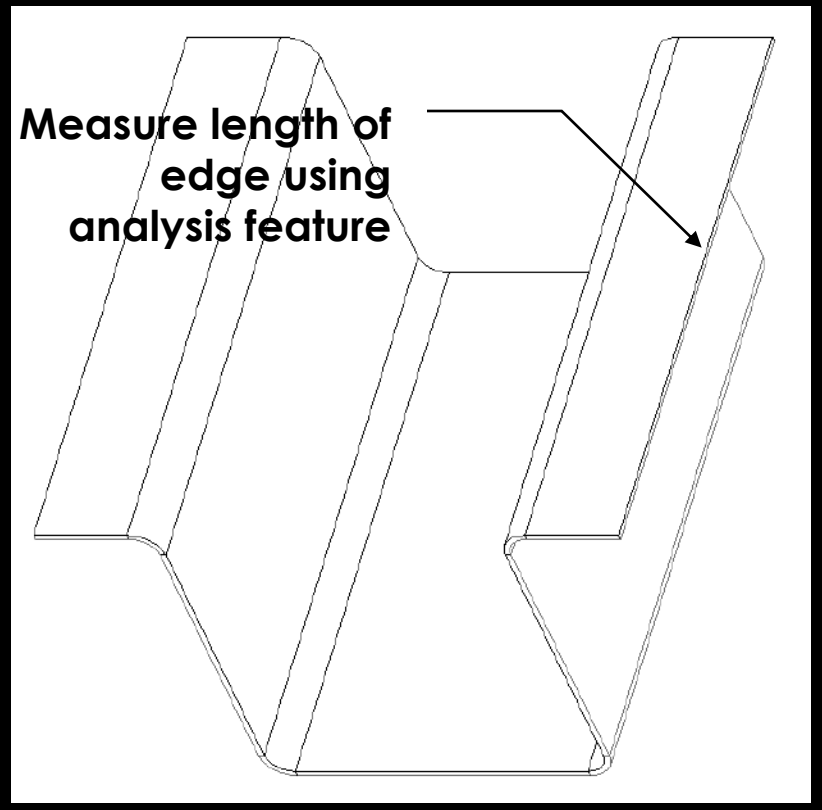
- Carefully design the UDF
  - Capture the full intent (not just the shape)
    - Think about how you want it to change in various situations.
  - Build “extra” features that help capture intent and include them in the UDF – analysis features, curves, surfaces

EXAMPLE - HOLES IN A PLATE

Objective: Create a pattern of equally-spaced holes on a flange.

Emphasis: capture the intent

1. Create an analysis feature to measure edge of part.

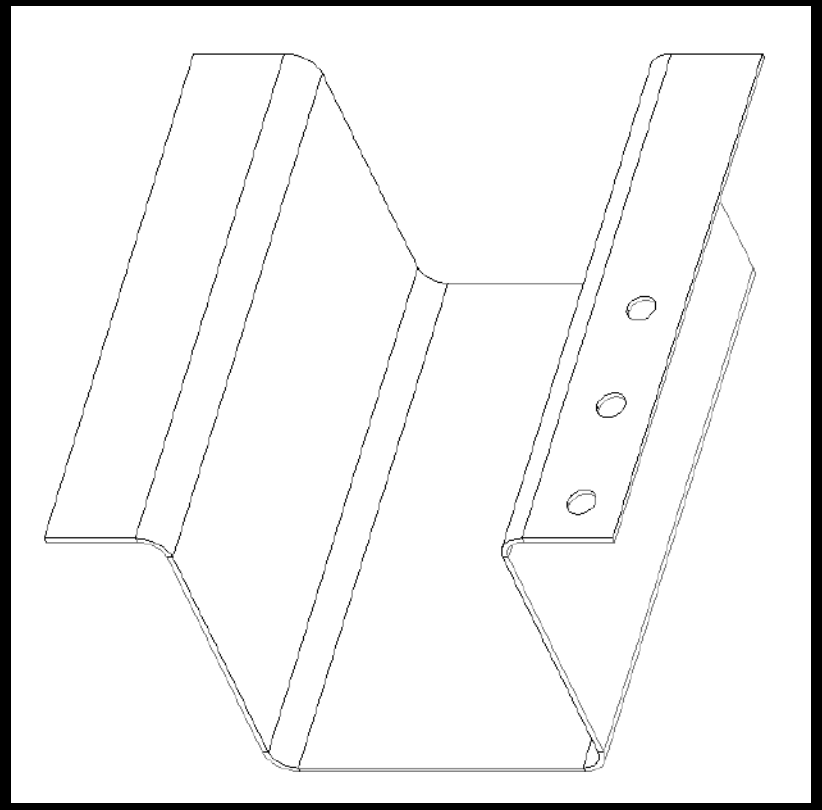


EXAMPLE - HOLES IN A PLATE

Objective: Create a pattern of equally-spaced holes on a flange.

Emphasis: capture the intent

1. Create an analysis feature to measure edge of part.
2. Create pattern of holes.

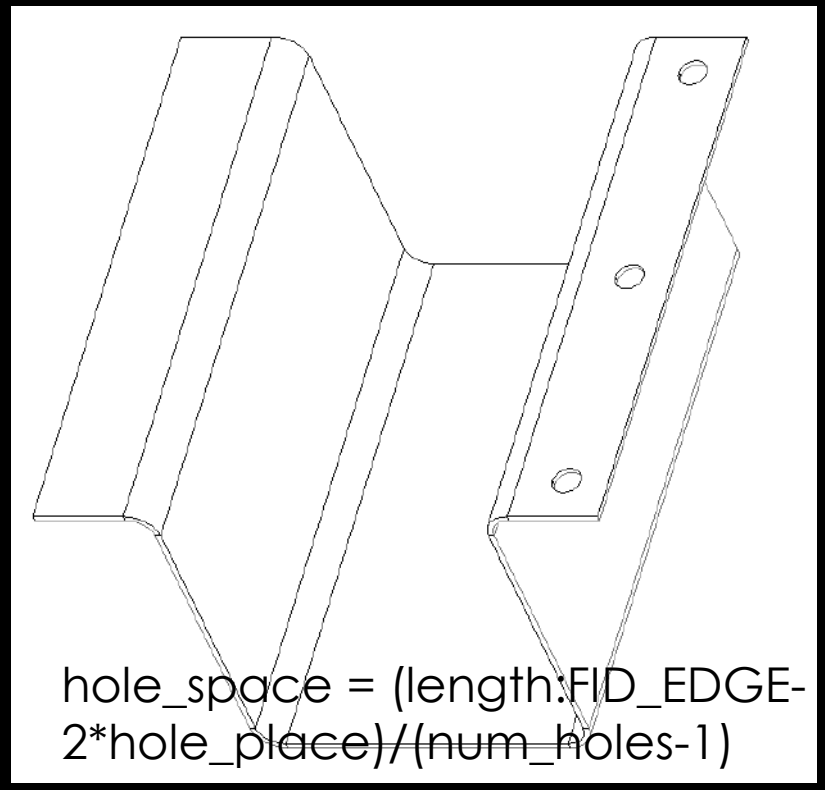


EXAMPLE - HOLES IN A PLATE

Objective: Create a pattern of equally-spaced holes on a flange.

Emphasis: capture the intent

1. Create an analysis feature to measure edge of part.
2. Create pattern of holes.
3. Write feature relation in first hole feature
4. Create UDF – analysis feature and holes

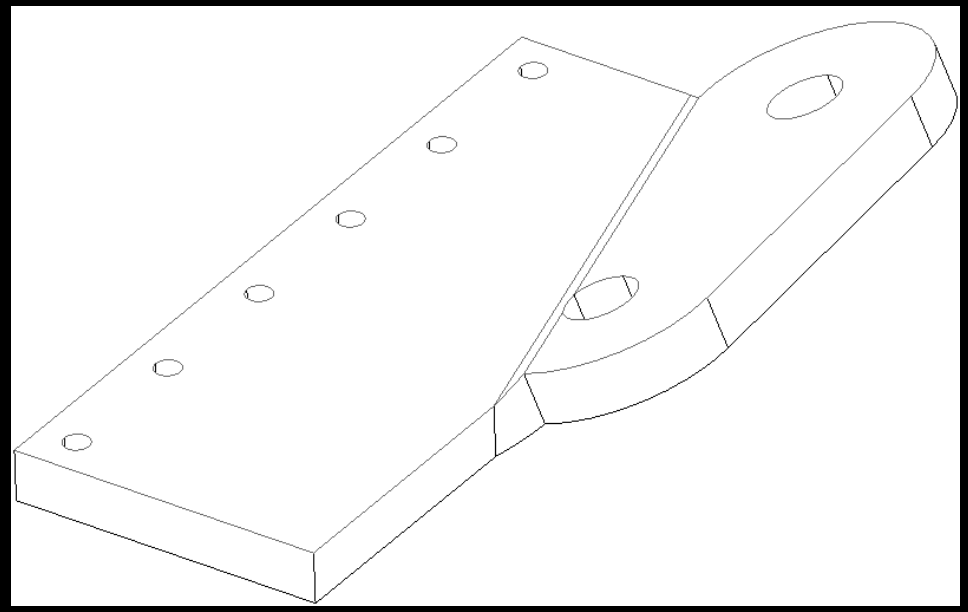


EXAMPLE - HOLES IN A PLATE

Objective: Create a pattern of equally-spaced holes on a flange.

Emphasis: capture the intent

1. Create an analysis feature to measure edge of part.
2. Create pattern of holes.
3. Write feature relation in first hole feature
4. Create UDF – analysis feature and holes
5. Use UDF – Modify dimensions as needed



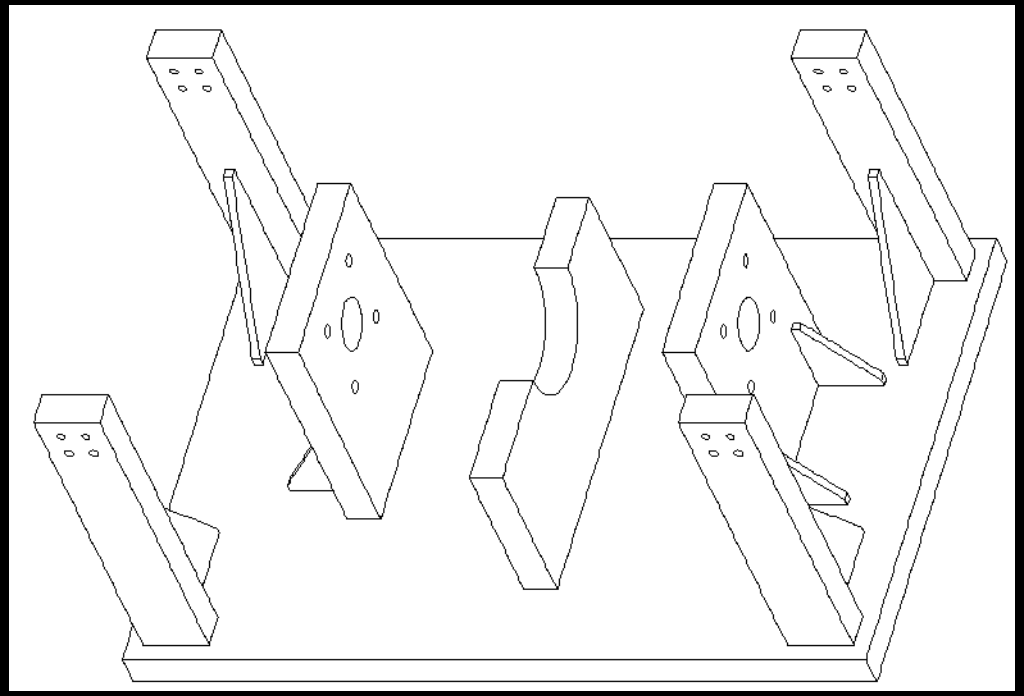
EXAMPLE – FIXTURE COMPONENTS AS FEATURES

Objective: Make a fixture weldment using features instead of parts

Emphasis: Be creative

Create rectangular extruded protrusion in UDF part.

- In protrusion, create feature parameters for length, width, height.
- Write feature relations making parameters equal their dimensions, e.g. length=d5
- Write notes or use Toolkit to make a table to query for the parameters for each feature

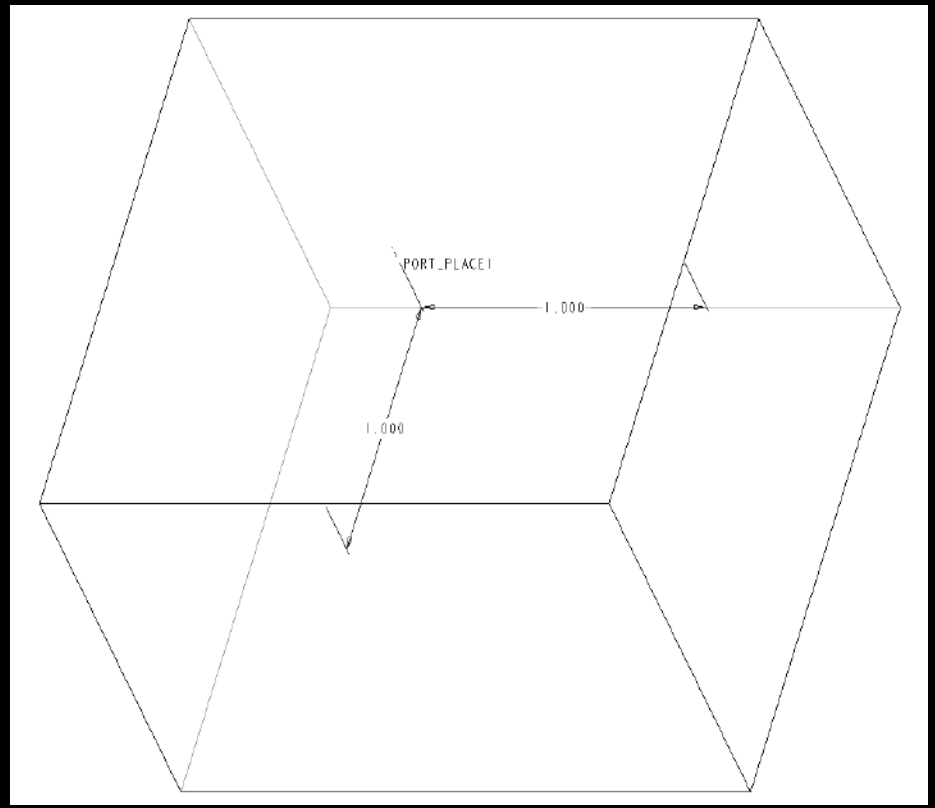


EXAMPLE – HYDRAULIC PORTS

Objective: Reusable hydraulic ports that adhere to standards

Emphasis: designing UDFs

1. Create an axis in the part.

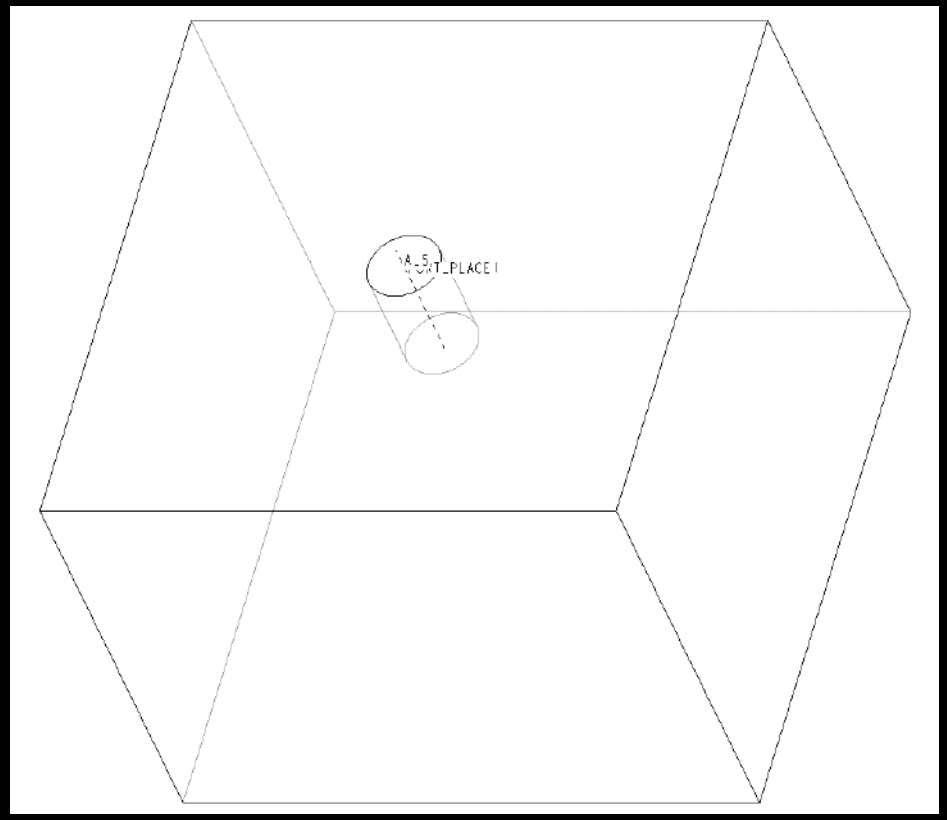


EXAMPLE – HYDRAULIC PORTS

Objective: Reusable hydraulic ports that adhere to standards

Emphasis: designing UDFs

1. Create an axis in the part.
2. Create a coaxial hole through the axis, placed on top face of protrusion.

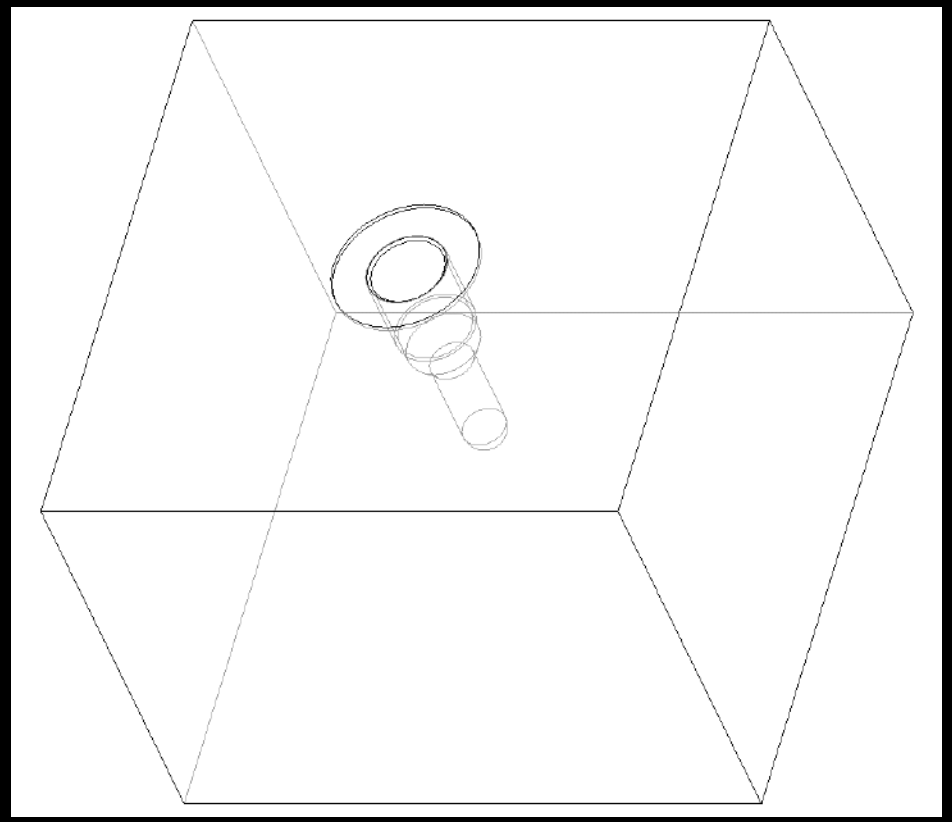


EXAMPLE – HYDRAULIC PORTS

Objective: Reusable hydraulic ports that adhere to standards

Emphasis: designing UDFs

1. Create an axis in the part.
2. Create a coaxial hole through the axis, placed on top face of protrusion.
3. Create other holes, chamfers, etc. to complete the special feature.
4. In the first hole feature, create some standard feature parameters for the port spec. Write feature relations to relate them as appropriate to make them follow feature dimensions.
5. Create the UDF, selecting the holes (not the axis)
6. Create a family table using different dimensions and features.
7. Optional: create a note file that reads the feature parameters from the first hole. Use "<feature parameter>:att"

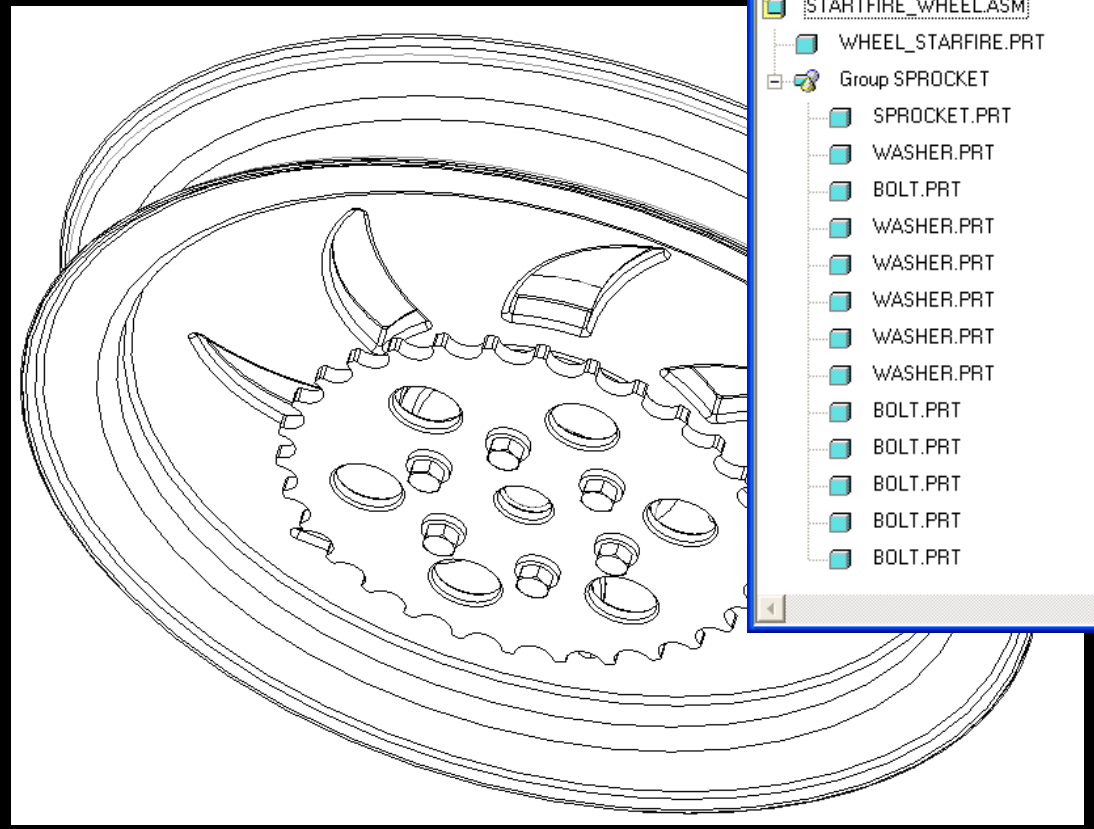


EXAMPLE - ASSEMBLY COMPONENTS

Parts/assemblies that go together but are not a subassembly.

Emphasis: UDF's not just for design parts

1. Create a new assembly for defining the UDF
2. Create simple parts for references for the UDF parts. A single part would be best.
3. Assemble the parts that will be part of the UDF.
4. Define the UDF.
5. To assemble all components, just pick assembly references (not component references)

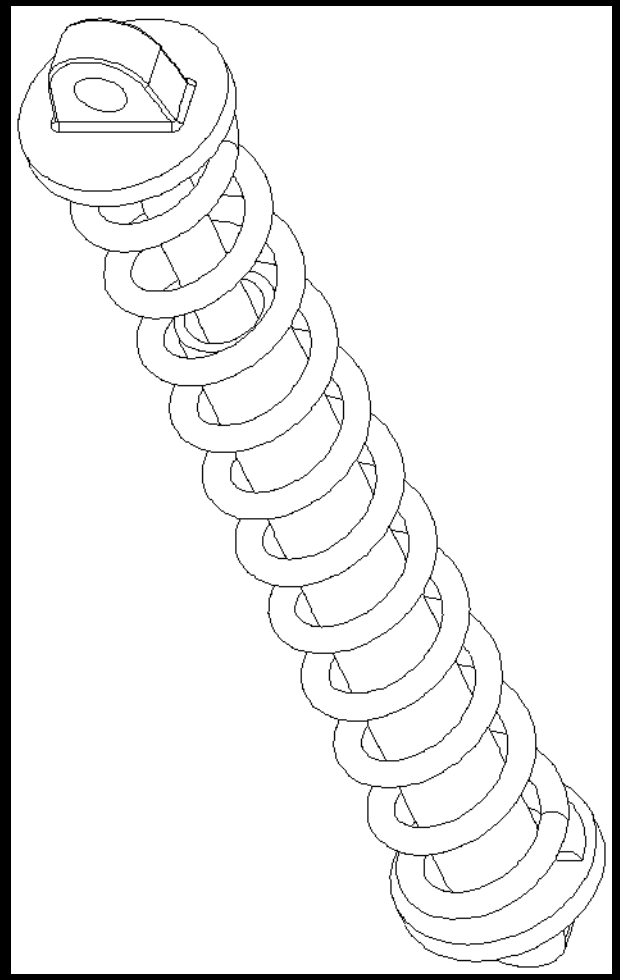


EXAMPLE – ASSY-VARIABLE SPRING

Objective – create a spring that will change sizes depending on the assembly where it's used.

Emphasis: Powerful Technology

1. Create a new part from start part.
2. Create offset datum plane from default.
3. Create analysis feature to measure the distance between offset plane and parent plane
4. Create spring features referencing the datum planes for length.
5. Write feature relation controlling spring pitch based on plane distance.
6. Define UDF – do not include planes.
7. Use by assembling a new part into an assembly using the default plane of part. Create feature in part, referencing an assembly plane for top end of spring.



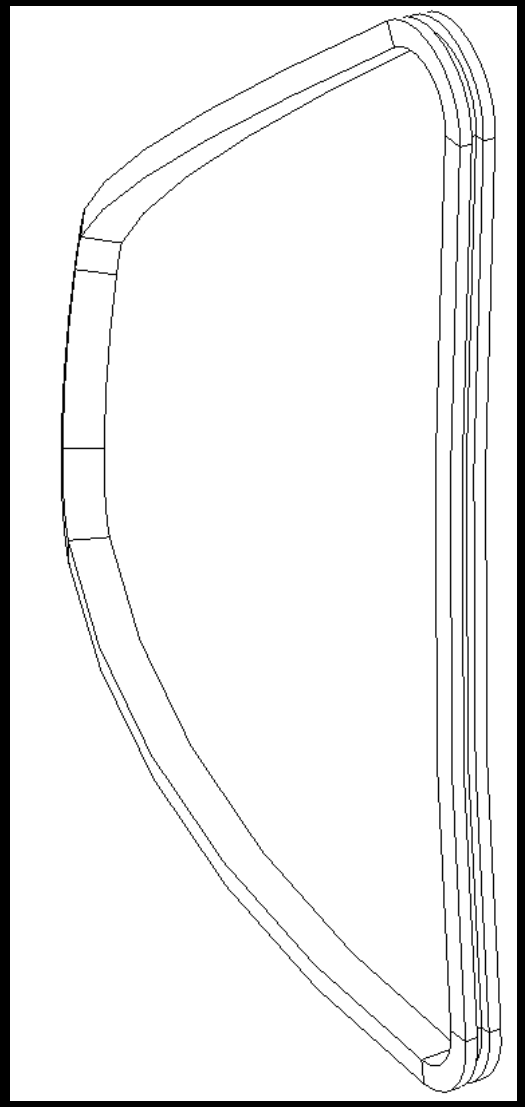
Shock_spring
Shock_absorber.asm

EXAMPLE – ASSY-VARIABLE SEAL

Objective – create a seal part that consists of standard material wrapped around a window or door edge.

Emphasis: Intent, not geometry

1. Create a new part
2. Create two parallel straight line curves next to each other.
3. Create a variable section sweep protrusion using one curve for the spine and one for the x-vector. Dimension/constrain section only to crosshairs, not geometry.
4. Create analysis feature measuring length of swept edge
5. Create UDF - include swept protrusion and analysis feature.
6. In assembly where used, make two curves that follow the contour of the door/window.
7. Copy geom the curves into a new part.



Create Def_seal
Car_window

OTHER EXAMPLES

- NC features in Mfg models
  - probing
  - port drilling operations (drill/ream/tap/spot face)
  - Expert Machinist is basically a library of UDF's
- Power tool motor housings
- Skeleton features – engine cylinder
- Mold features – sprues, runners, gates
- User-defined analyses for behavioral modeling
- Pipes & hoses
- Hole creation interface is UDF with software interface added
- Sheet metal punches and notches

SUMMARY

- UDF's capture your unique design intent in custom features
- UDF's are powerful tools that can be used to make work quicker in many areas by turning complex features/components into pick and place features