

Programming in Pro/ENGINEER Wildfire 2.0

January 18, 2005

1 Introduction

Pro/ENGINEER supports interactive graphical programming at two different levels. At the higher level, C++ programs are supported through Pro/ENGINEER API Toolkit. At lower level, a micro programming environment, Pro/E PROGRAM Tool, is supported. These programming environments serve different needs. In this document, we will discuss the background of the Pro/E API Toolkit, while focus on the Pro/E Program Tool due to its ease of use.

1.1 Pro/ENGINEER API Toolkit

Pro/ENGINEER API Toolkit allows customers to extend, automate, and customize a wide range of Pro/ENGINEER design-through-manufacturing functionality.

Pro/ENGINEER API Toolkit consists of a library of functions:

- an application-programming interface (API), written in the C programming language. These functions are typically used by MIS organizations to create applications that run in parallel with Pro/ENGINEER and to integrate product information with the customer's corporate MRP/ERP systems.
- applications used extensively by companies participating in PTC's Cooperative Software Partner (CSP) program to interface their commercial information management products with Pro/INTRALINK. Normally, participation of a three day tutorial on the API Toolkit is needed to get the API Toolkit function module.

The extensive Pro/ENGINEER API Toolkit provides programmatic access for creating, interrogating, and manipulating almost every aspect of the engineering model and its data management.

Typical toolkit applications include:

- automating the creation of complex features
- automating the production of Pro/ENGINEER deliverables, such as BOMs, drawings, and manufacturing operations
- improving product quality by performing design rule verification based on inputs from an external, knowledge-based system.

More specifically, this Pro/ENGINEER API Toolkit functionality allows:

- Customization of the Pro/ENGINEER-Foundation menu system
- Datum, solid, and manufacturing feature creation
- Assemblies
- Drawing automation
- Access to model geometry

The Pro/ENGINEER API Toolkit provides complete access to the information within the Pro/INTRALINK environment, allowing customers to further leverage the product information contained within Pro/INTRALINK.

Specifically, this functionality allows:

- Integration with MRP/ERP Systems
- Custom client applications, such as Web integrated clients
- Triggered verification, notification and enforcement of business process actions

Product Capabilities:

- Create automated, single-use or derived designs by geometric and parametric constraints
- Extend the Pro/ENGINEER user interface with custom processes seamlessly embedded in the interface
- Customize the Pro/ENGINEER menu system
- Collaborate between Pro/ENGINEER applications
- Access peer-to-peer communications for better application diagnoses

Customer Benefits:

- Integrate expert systems and knowledge-based applications into the Pro/ENGINEER environment.

Improve product quality with design rule verification based on inputs from an external, knowledge-based systems.

1.2 Pro/E Program Tool

The Pro/E Program environment, on the other hand, support quick and relatively straightforward interactive graphical programming in Pro/E for every users.

The programming environment is simply Pro/E and *Microsoft Notepad* or *Word*.

One can enter the Pro/E PROGRAM environment, by clicking **Tools > Program...** from the pull-down menu in the Pro/E **PART** or **ASSEMBLY** mode.

To show or edit the program, one can click **Show Design** or **Edit Design** from the **PROGRAM** menu.

A typical Pro/E PROGRAM routine may contain any of the following:

- Input variables
- Relations
- IF-ELSE clauses
- Lists of all the features, and parts
- INTERACT statements
- MASSPROP statement

After the Pro/E PROGRAM routine is edited, the user will be asked whether the changes are to be incorporated (in the message window at the bottom). To proceed, enter Y. If N entered, the program will not be executed and changes will be lost.

2 Programming Details

The ingredients of a typical program include:

2.1 Input Variables

INPUT variables may be specified at the beginning of the listing; and their values can be provided by the user at the program beginning.

The format of INPUT is as follows:

```
INPUT
Variable_Name Variable_Type
END INPUT
```

The INPUT statement must define the name and type of the variable. Variable names must always begin with a character. The following variable types are supported:

Number

String: This enables the user to enter parameters or model names.

Logical (YES_NO): Enter either Y or N.

An example:

```
INPUT
THICKNESS NUMBER
"Enter wall thickness for the cylinder"
END INPUT
```

2.2 Relations

All valid relations in a Pro/ENGINEER model can be entered in a Pro/PROGRAM.

An example:

```
d0 = d6 * 2
```

Here, d0 and d6 are dimension ID name.

2.3 IF-ELSE Clauses

Conditional statements, i.e. IF _ ELSE, can be used to create a program branch.

For example:

```
ADD PROTRUSION.....
IF d1 > d2
ADD HOLE ...
END ADD
ENDIF
ADD CUT.....
END ADD
```

So, when d1 is smaller than d2, a CUT is added, instead of a HOLE.

2.4 Lists of Features and Parts

The program that Pro/E PROGRAM brings up simply includes all feature building commands used in creating the model and the properties of these features. All features and parts are listed in the program. For instance, the ADD feature by EXTRUSION operation is recorded as:

```

ADD FEATURE (initial number 8)
INTERNAL FEATURE ID 106
PARENTS = 100(#7)
PROTRUSION: Extrude

```

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Add
4	Section	Defined
4.1	Reference Sketch	F7(SKETCH_2)
5	Feature Form	Solid
6	Direction	Side 2
7	Depth	Defined
7.1	Side One	Defined
7.1.1	Side One Depth	None
7.2	Side Two	Defined
7.2.1	Side Two Depth	Variable
7.2.2	Value	70.00

```
SECTION NAME = Sketch 2
```

```

FEATURE'S DIMENSIONS:
d11 = 70.00
END ADD

```

Additional operations can be added, and this ADD operation can be changed.

2.5 INTERACT

INTERACT statements provide a placeholder for creating interactive part. They can be inserted anywhere within the FEATURE ADD - END ADD.

Here is an example,

```

ADD PROTRUSION.....
IF d1 > d2
ADD HOLE.....
ELSE
INTERACT
END IF
ADD CUT.....

```

In this example, an alternate set of features will be created if d1 is not greater than d2. The ADD CUT command has to be input by the user.

2.6 MASSPROP

The MASSPROP statement is used to update mass properties each time geometry changes. Format is as follows:

```

MASSPROP
END MASSPROP

```

2.7 Other Operations for Feature Editing

a) Changing feature dimension

The dimensions of features in the program can be updated by a DIMENSION statement with:

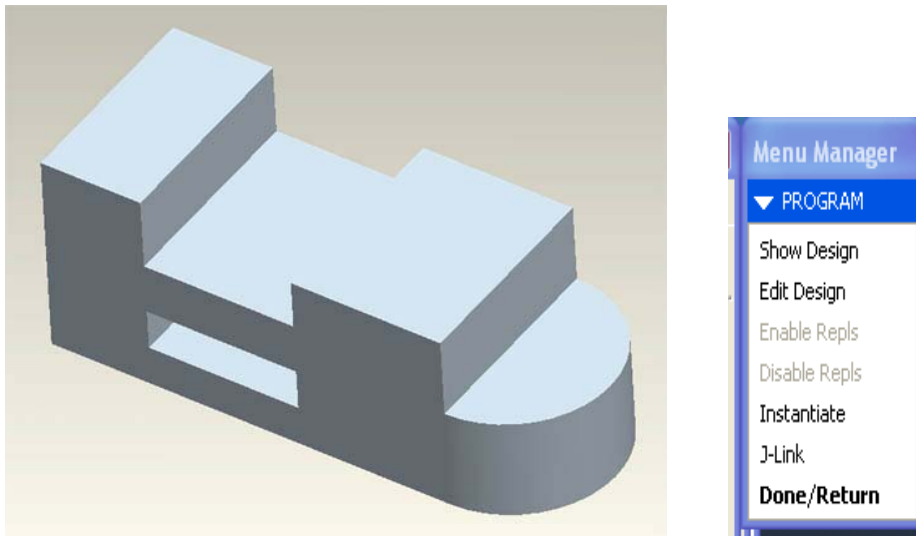
```
MODIFY d# = value
```

b) Editing Errors

Common editing errors include:

- Having an IF statement without an END IF statement or vice versa
- Typing a variable name incorrectly in a relation or a condition
- Reordering a child before the parent
- Deleting a parent feature

An Example Part and the PROGRAM Window



The Pro/Program for this Part Model

- Start Pro/E
- Open the Part Model file: **part5.prt**
- Use Pull Down Menu **T**ool > **P**rogram...
- In the **PROGRAM** Window
 - **Show Design** and **Edit Design** options will display the Pro/Program that is used to create the displayed part model.
 - **Edit Design** option allows you make changes to the model through “Programming Logic” rather than through “drawing and modeling”. Automated tasks can be achieved. If you exit from the Edit window and answer “Yes” in the message window at the bottom of the screen to the prompt: “Do you want to incorporate your changes into the model?” The programmed change will be added to the existing model. You can start from a simple template model to write various programs.
 - The **J-Link** function allows you to load in Java codes.

The List of the Pro/E PROGRAM for this Part Model

VERSION 2.0
 REVNUM 365
 LISTING FOR PART LESSON5

INPUT
 END INPUT

RELATIONS
 END RELATIONS

ADD FEATURE (initial number 1)
 INTERNAL FEATURE ID 1

DATUM PLANE

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Constraints	Defined
2.1	Constraint #1	Defined
2.1.1	Constr Type	X Axis
3	Flip Datum Dir	Defined
4	Fit	Defined
4.1	Fit Type	Default

NAME = RIGHT

END ADD

ADD FEATURE (initial number 2)
 INTERNAL FEATURE ID 3

DATUM PLANE

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Constraints	Defined
2.1	Constraint #1	Defined
2.1.1	Constr Type	Y Axis
3	Flip Datum Dir	Defined
4	Fit	Defined
4.1	Fit Type	Default

NAME = TOP

END ADD

ADD FEATURE (initial number 3)
 INTERNAL FEATURE ID 5

DATUM PLANE

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Constraints	Defined
2.1	Constraint #1	Defined
2.1.1	Constr Type	Z Axis

3 Flip Datum Dir Defined
 4 Fit Defined
 4.1 Fit Type Default

NAME = FRONT

END ADD

ADD FEATURE (initial number 4)
 INTERNAL FEATURE ID 7
 PARENTS = 1(#1) 3(#2) 5(#3)

PROTRUSION: Extrude

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Add
4	Section	Defined
4.1	Setup Plane	Defined
4.1.1	Sketching Plane	FRONT:F3(DATUM PLANE)
4.1.2	View Direction	Side 1
4.1.3	Orientation	Top
4.1.4	Reference	TOP:F2(DATUM PLANE)
4.2	Sketch	Defined
5	Feature Form	Solid
6	Direction	Side 2
7	Depth	Defined
7.1	Side One	Defined
7.1.1	Side One Depth	None
7.2	Side Two	Defined
7.2.1	Side Two Depth	Variable
7.2.2	Value	10.00

NAME = BLOCK
 SECTION NAME = S2D0001

FEATURE'S DIMENSIONS:
 d2 = 20.00
 d3 = 10.00
 d4 = 10.00
 END ADD

ADD FEATURE (initial number 5)
 INTERNAL FEATURE ID 28
 PARENTS = 7(#4)

PROTRUSION: Extrude

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Add
4	Section	Defined
4.1	Setup Plane	Defined
4.1.1	Sketching Plane	Surf:F4(PROTRUSION)
4.1.2	View Direction	Defined

4.1.3 Orientation Right
 4.1.4 Reference Surf:F4(PROTRUSION)
 4.2 Sketch Defined
 5 Feature Form Solid
 6 Material Side Side Two
 7 Direction Side 2
 8 Depth Defined
 8.1 Side One Defined
 8.1.1 Side One Depth None
 8.2 Side Two Defined
 8.2.1 Side Two Depth Variable
 8.2.2 Value 5.00

NAME = ROUND_END
 SECTION NAME = S2D0002
 OPEN SECTION

FEATURE'S DIMENSIONS:
 d9 = 5.00
 END ADD

ADD FEATURE (initial number 6)
 INTERNAL FEATURE ID 52
 PARENTS = 7(#4) 28(#5)

CUT: Extrude

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Remove
4	Section	Defined
4.1	Setup Plane	Defined
4.1.1	Sketching Plane	Surf:F4(PROTRUSION)
4.1.2	View Direction	Side 1
4.1.3	Orientation	Top
4.1.4	Reference	Surf:F5(PROTRUSION)
4.2	Sketch	Defined
5	Feature Form	Solid
6	Material Side	Side Two
7	Direction	Side 1
8	Depth	Defined
8.1	Side One	Defined
8.1.1	Side One Depth	None
8.2	Side Two	Defined
8.2.1	Side Two Depth	Variable
8.2.2	Value	10.00

NAME = TOP_CUT
 SECTION NAME = S2D0003
 OPEN SECTION

FEATURE'S DIMENSIONS:
 d12 = 5.00
 d13 = 7.50
 d14 = 3.50
 d15 = 2.40
 d16 = 10.00
 END ADD

ADD FEATURE (initial number 7)

INTERNAL FEATURE ID 149
 PARENTS = 3(#2) 52(#6) 7(#4)

CUT: Extrude

NO.	ELEMENT NAME	INFO
1	Feature Name	Defined
2	Extrude Feat type	Solid
3	Material	Remove
4	Section	Defined
4.1	Setup Plane	Defined
4.1.1	Sketching Plane	Surf:F4(PROTRUSION)
4.1.2	View Direction	Side 1
4.1.3	Orientation	Top
4.1.4	Reference	Surf:F6(CUT)
4.2	Sketch	Defined
5	Feature Form	Solid
6	Material Side	Side Two
7	Direction	Side 1
8	Depth	Defined
8.1	Side One	Defined
8.1.1	Side One Depth	None
8.2	Side Two	Defined
8.2.1	Side Two Depth	Thru All

NAME = INSIDE_CUT
 SECTION NAME = S2D0001

FEATURE'S DIMENSIONS:
 d24 = 2.00
 d25 = 2.00
 END ADD

MASSPROP
 END MASSPRO

