SAND2007-3742P

Cubit User Meeting

Using Aprepro In CUBIT

Darryl Melander

LOCKHEED MARTIN

Sandia National Laboratories

Outline

- Introduction to Aprepro
- Variables
- Operators
- Functions
- Control Statements
- Examples

Introduction to Aprepro

- **Aprepro = Algebraic Pre-Processor**
- **A built-in miniature programming language**

- **Purposes**
  - Parameterize a journal file
  - Error checking
  - Other control logic

Basic Aprepro Syntax

- **Aprepro expressions are wrapped in curly braces**
- **Aprepro evaluated first, results inserted into command:**

```
Brick X {10}
```

And

```
Brick X {5*2}
```

Are Equivalent To

```
Brick X 10
```

Aprepro Variables

- **Variables are named values**
- **Names are case sensitive**
- **Variable type is Number or String**
- **Defined in a comment**

```
# {width=2}
# {r="radius"}
```

- **Use anywhere in a command, as if you typed the variable's value:**

```
brick x {width}
cylinder height 5 {r} 10
```

Aprepro Equations

- **Variable values can be changed**

```
# {x = 1}
# {x = 5}  ##This will give you a warning
# {x++}  ## Increase value of x by one, no warning
```

  - To avoid warnings, make variable name start with an underscore (_), or use ++ and --

- **Convenient way to see variable value: *comment* command**

```
Comment x
User Comment: 6

Comment "x is" x "and y is" y
User Comment: x is 6 and y is <undefined>
```

Operators

- **Addition: +**
- **Subtraction: -**
- **Multiplication: ***
- **Division: /**
- **Power of: ^ ({3^2} = 9)**

- **Math expressions can be used just about anywhere:**

```
# {width=3}
# {width_squared = width^2}
brick x {width} y {width*2} z {width_squared}
```

Aprepro Functions

- **Functions calculate or look up values**
- **Function name followed by parameters in parentheses**
- **The number and type of parameters depends on the function**
- **Commas between parameters**
- **Parameters can be constants, variables, or equations**

```
# {errs = get_error_count()}
# {x=cos(PI/2)}
# {vertex_x = Vx(30)}
# {random_number = rand(10, 20)}
# {Print ("Hello World")}
```

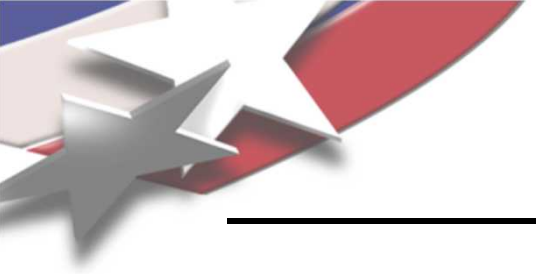
Types of Functions

- **Math Functions**
  - sin(num), cos(num), asin(num), etc…
  - sqrt(num), exp(num), log(num), ln(num), etc…
- **String Manipulation Functions**
  - Quote(string), toupper(string), tolower(string)
- **Utility Functions**
  - Print(string), PrintError(string)
  - FileExists(string), HasFeature(string)

Types of Functions

- **Session Information Functions**
  - NumVolumes(), NumSurfaces(), etc…
  - get_error_count(), set_error_count(num)

- **Entity Information Functions**
  - Vx(num), Nz(num)
  - Radius(num), SurfaceArea(num), Length(num)
  - CurveAt(num, num, num), HexAt(num, num, num)



Flow Control

- **Three types of flow control**
  - If statements
  - Loops
  - Include files

If Statements

- **If true then do**

```
#{if (my_variable < 3)}
  brick x 3
#{else}
  brick x {my_variable}
#{endif}
```

- **If defined and not zero then do**

```
#{ifdef(make_a_brick)}
  brick x 3
#{endif}
```

- **If zero or not defined then do**

```
#{ifndef(skip_the_brick)}
  brick x 3
#{endif}
```

Loops

- **Repeat a set of commands some number of times**

```
# Create 3 bricks
#{Loop(3)}
  brick x 10
#{EndLoop}
```

- **Loop statement accepts numbers or variables, but not expressions**

```
# {Loop(3)}  #OK
# {Loop(x)}  #OK
# {Loop(x-1)}  #error
```

Include Files

- **Include statement "pastes" contents of another file into journal file**
- **A lot like the CUBIT "playback" command**

  # {include ("filename")}

Example 1 - Parameterized Sizes


```
# {geom_size=3}
# {mesh_size=.5}
Brick x {geom_size}
Volume 1 size {mesh_size}
Mesh Volume 1
```

Example 2 – Check for Success


```
#{errors_before=get_error_count()}
Blah blah blah
#{If(get_error_count() > errors_before)}
  #{PrintError("Woops")
  Quit
#{EndIf}
```

Example 3 – Accurate Coordinates

Create Vertex {Vx(1) + 10} {Vy(1)} {Vz(1)}

Example 4 – Count Entities


```
#{Loop(5)}
  Brick x 10
#{EndLoop}
Mesh Volume All
Group "all_meshed_vols" add volume with is_meshed = true
#{If(NumInGrp("all_meshed_vols") == NumVolumes())}
  Comment "Hurray, everything meshed"
#{Else}
  #{PrintError("Some of the volumes didn't mesh!")}
  quit
#{EndIf}
```

Example 5 – Name that Volume

Create Brick Width 1
Create Brick Width 1
Volume {Id("volume")-1} Name "Martha"
Volume {Id("volume")} Name "George"

Example 6 – Partless Volumes


```
Import acis 'example6.sat' xml 'artifact.dta'
# Loop through the volumes
# {num_vols=NumVolumes()}
# {cur_id = 1}
# {original_err_count = get_error_count()}
# {cur_err_count = original_err_count}
# {ids_string = "Volumes without metadata: "}
# {Loop(num_vols)}
# {PartInVol(cur_id)}
# {If(get_error_count() != cur_err_count)}
# {cur_err_count = get_error_count()}
# {ids_string = ids_string // tostring(cur_id) // " "}
# {EndIf}
# {cur_id++}
# {EndLoop}
# {set_error_count(original_err_count)}
# Print out the results
Comment ids_string
```

Example 7 – Row Of Bricks


```
# ***Set parameters***
# {num_bricks=5}
# {brick_size=1}
#
# ***Create the bricks***
# {Loop(num_bricks)}
  Brick Width {brick_size}
# {EndLoop}
#
# ***Scoot them into a line***
# {cur_brick = 1}
# {Loop(num_bricks)}
  Volume {cur_brick} move {(cur_brick-1)*brick_size}
  #{cur_brick++}
# {EndLoop}
```

Example 8 – Connect the Dots


```
# ***Create Points***
Create Vertex 0 0 0
Create Vertex 1 0 0
Create Vertex .5 1 0
#
# ***Create id string***
# {_id_str = " "}
# {vert_count = NumVertices()}
# {cur_vert_id=1}
# {Loop(vert_count)}
  # {_id_str = _id_str // " " // tostring(cur_vert_id++)}
# {EndLoop}
#
# ***Create the Surface***
Create Surface Vertex {_id_str}
```

Example 9 – Working with Files

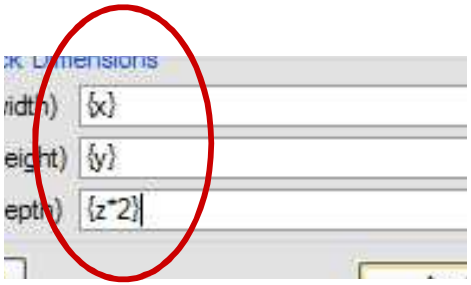

```
# {myfile = “foo.jou”}
Play {Quote(myfile)}



#Play f1.jou to f3.jou
# {cur=1}
# {Loop(3)}
  play {Quote(“f” // tostring(cur++) // “.jou”)}
# {EndLoop}
```

Aprepro in the GUI

- **Use aprepro anywhere in the GUI, using curly braces**



For More Information

- **Online Users Manual: http://cubit.sandia.gov**
  - Click on “Documentation”
  - Click on “CUBIT 10.2 On-Line User's Manual (October 2006)”
  - Click on “Appendix” in Table of Contents
  - Click on “APREPRO”