FeneVision® Best Practice

Fixing Shape Parameter Attributes

Introduction

There was an error in the shape scripts in an early version of the template database where the attribute scripts incorrectly assume SP-W is the ordered width. This is true for most shapes, but there are some where this is always incorrect. There is also a subset of shapes where SP-W is the ordered width but can be different based on the shape parameters.

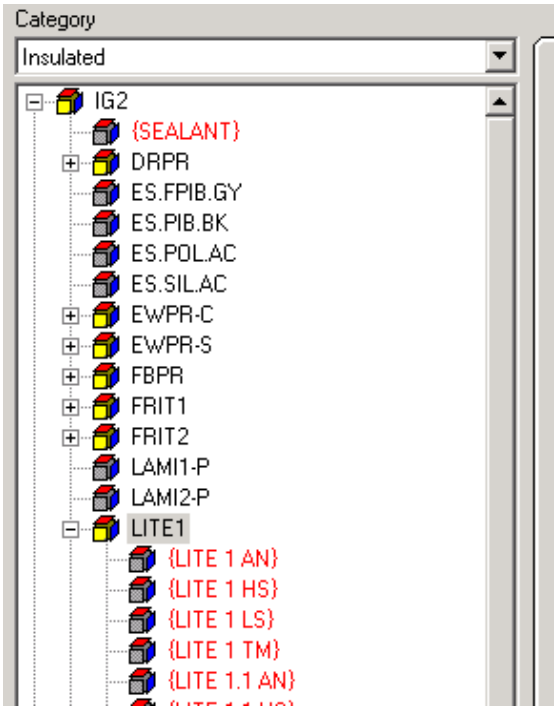
This document will outline the steps needed to correct the attribute scripts. New attributes will need to be added so that the system can differentiate shape width (SP-W attribute) from ordered width (W attribute)

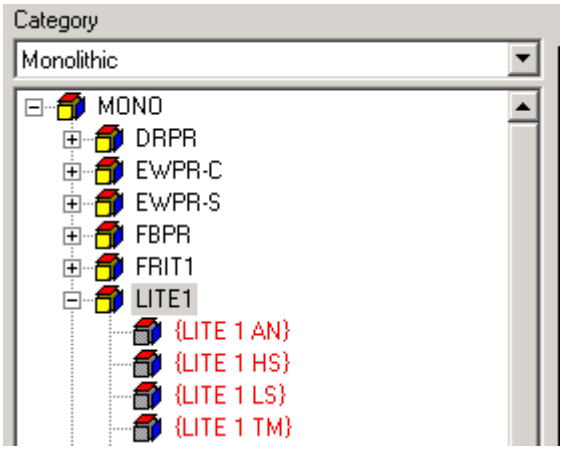
*** For readability, the document will reference W and SP-W. Anything you do to W or SP-W should be duplicated for H and SP-H unless otherwise noted.

Assumptions

This document assumes the dynamic glass parts have all the SP- attributes, but do not have H or W attributes.

This document will assume the BOM is similar to what is shown below.





Using

To fix the issue attribute scripts will need to be modified and attributes will need to be added.

1. Update 10-SHPINIT

10-SHPINIT will need to be modified to track SP-W and the ordered width as separate variables. We will use the existing W attribute to contain SP-W and create a new variable called WF to hold the ordered width.

1. Obtain a copy of 10-SHPINIT from the current template database, attached to this article.
2. Use a diff program to show the differences between the 10-SHIPINIT attribute script from the latest template database and your own version of the script. There will be a lot of differences; just focus on the differences that use the W and WF variables.

2. Update all other 10-SHP* scripts

There may be additional 10-SHP* attributes, such as 10-SHPADJ and 10-SHPADJ-SPACER. Review those. They should be somewhat similar to 10-SHPINIT. Update those scripts to use W and WF as needed.

3. Update the Shape options to require W and H values

The user will need to specify the SP-W value in the options wizard. To configure this, the shape option must be updated in Parts Setup.

1. In Parts Setup to go the Options Tab and find the Shape option (it is usually a System option)
2. Use the table below to identify which shapes require an update. (Example, Shape 47)
3. Change the Default Value box to include “H = “ or “W = “ as indicated in the table. (Shape 47 will require W)
4. When all shapes have been updated select the Shape Library question. Uncheck “Require User Input Value” and recheck it (to enable the Commit button). Click the commit button.

Shape Number	Add W?	Add H?	Notes
21	Y		
22	Y		
24	Y		
47	Y		
48	Y		
66	Y	Y	If R is too large the curved edge will extend beyond the H or W.
67	Y	Y	If R is too large the curved edge will extend beyond the H or W.
68		Y	If R is too large the curved edge will extend beyond the H or W.
69		Y	If R is too large the curved edge will extend beyond the H or W.
70	Y		If R is too large the curved edge will extend beyond the H or W.
71	Y		If R is too large the curved edge will extend beyond the H or W.
72	Y	Y	If R is too large the curved edge will extend beyond the H or W.
73	Y	Y	If R is too large the curved edge will extend beyond the H or W.
76		Y	
77		Y	
152	Y	Y	If R is too large the curved edge will extend beyond the H or W.
153	Y	Y	If R is too large the curved edge will extend beyond the H or W.
154	Y	Y	If R is too large the curved edge will extend beyond the H or W.
155	Y	Y	If R is too large the curved edge will extend beyond the H or W.
330		Y	
331		Y	

4. Update the attributes on the ordered part

The ordered part should already have H and W attributes. We need to make sure the W and H attributes on the ordered part are the full height and width and not the values from SP-W / SP-H.

For example, the W attribute will have code similar to this:

```
ShapeDim = Attributes("SHPSTR").value
strpos = 0
colpos = 0
UnitEq = "W="
If InStr(ShapeDim, UnitEq) Then
        strpos = InStr(ShapeDim, UnitEq)
```

```
        colpos = InStr(strpos, ShapeDim, ";")
        retval = Mid(ShapeDim, strpos + Len(UnitEq), colpos - strpos - Len(UnitEq))
        retval = Val(retval)
End If
```

Update the script to pull the WF value from the shape string.

```
ShapeDim = Attributes("SHPSTR").value
strpos = 0
colpos = 0
UnitEq = "WF="
If InStr(ShapeDim, UnitEq) Then
        strpos = InStr(ShapeDim, UnitEq)
        colpos = InStr(strpos, ShapeDim, ";")
        retval = Mid(ShapeDim, strpos + Len(UnitEq), colpos - strpos - Len(UnitEq))
        retval = Val(retval)
End If
```

Update the H attribute as well, this time using “HF=”.

5. Add W and H attributes to the LiteX parts

The glass parts will need W and H attributes. These attributes should reference the W and H attributes of the parent.

6. Add W and H attributes to the dynamic glass parts

The dynamic glass parts will need W and H attributes. These attributes should reference the W and H attributes of the parent. At this point the glass parts know the ordered width and the value of the shape’s W attribute.

Testing

This modification should be done in a test server and not moved to live until tested. Units should be run through the entire system, paying special attention to:

1. Pricing
2. Paperwork / Reports
3. Opti