Volume

1

FENETECH, INC.

FeneVision® Configuration Guidelines

Users Manual

FeneVision® Configuration Guidelines
Users Manual

Fenetech, Inc.
1455 Danner Drive
Aurora, Ohio 44202
Phone 330-995-2830 • Fax 330-562-8688
www.fenetech.com

Table of Contents

Introduction

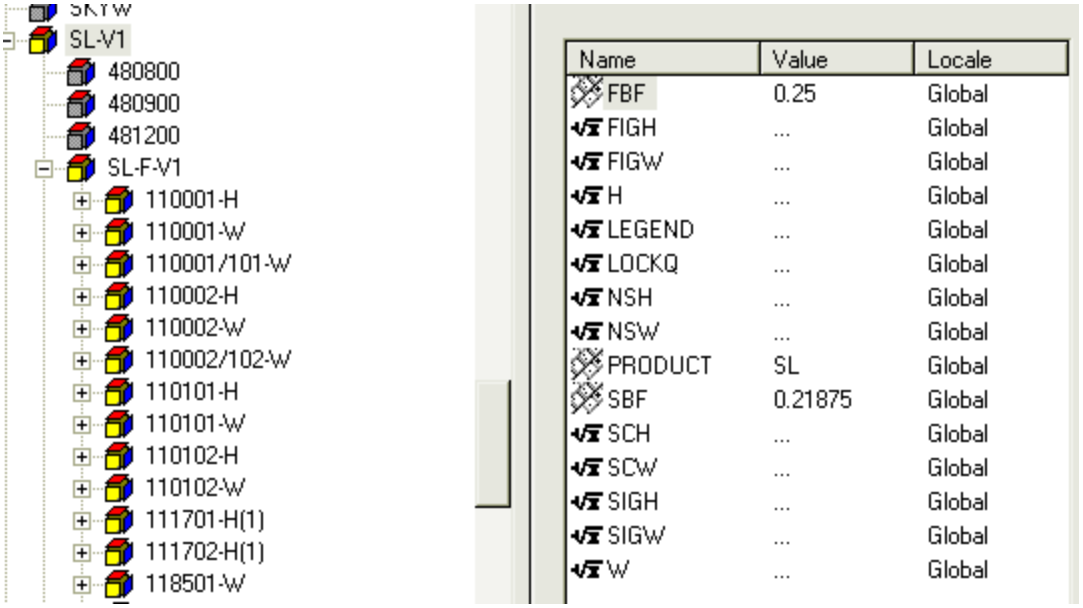
Welcome to FeneVision® Configuration Guidelines. This document is meant to assist in the creation of FeneVision® bill of materials configurations. The following guidelines assume the user is familiar with FeneVision® software and possesses basic configuration knowledge.

Configuration Structure

The configuration structure is many times the most important item to be concerned with when developing a bill of materials. Many interfaces and reports rely on the fact that the bill of materials structure is set up properly. If certain parts are structure properly they may be easily reused multiple times between different assemblies, and also within a single model.

Top Level Formulas:

It is helpful to create the most important cut logic formulas at the orderable part level. This is shown in the image below.



As can be seen, the topmost part is selected and there are many attributes assigned to this part. There are attributes for items such as; Fixed IG Height, Net Sash Height, Screen Height, and Burn Factor.

One of the reasons for doing this is so the attributes may be referenced by multiple sub-parts without recalculating anything, which is very useful when formulas need to be modified. This also organizes the formulas in one location so one does not have to search through multiple levels to locate cut logic formulas.

Glass Configuration

This document describes how to configure glass in FeneVision bill of materials. Many of the procedures and guidelines described require modification to best suit each manufacturers needs.

General Glass Rules:

The following rules may apply to Single Strength (3/32") glass:

- Cannot be Tempered
- Typically cannot be tinted or obscured.
- Can be used if size is less than 12 sqft.

The following rules may apply to Double Strength (1/8") glass:

- Lowest thickness for Tempering
- Used for sizes ranging from 12 sqft. to 22 sqft.

The following rules may apply to Triple Strength (3/16") glass:

- Used for sizes ranging from 22 sqft. to 35 sqft.

The following rules may apply to Quarter Inch (1/4") glass:

- Used for sizes greater than 35 sqft.

Options Structure:

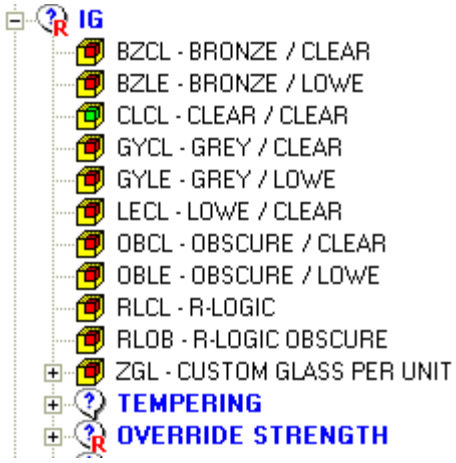


Image 1

Image 1 is an example of a commonly used options structure. Each IG unit available is listed using a smart code that describes the IG unit (BZ=bronze, CL=clear). Some manufacturers prefer to create these codes in inner/outer lite style, however some like to create codes so they appear by first letter so users in order entry can use the keyboard to jump to options (as shown above).

The "TEMPERING" question following the "IG" question is dependant on the way tempering is priced. If tempering is one standard charge for every type of glass then a tempering question can be used. If tempering prices are different between types of glass then separate option codes must be created for each tempered IG unit. For example: if each of the above IG option codes were offered in tempered then the user would create an option code for each, perhaps prefixed with a "T". This would double the amount of option codes available in the "IG" question.

Additional question regarding glass are found nested within the "IG" question. This is done so that if the order entry user were to pick "<none>" for the IG question, all the nested questions would be skipped. Some examples of additional questions are: "SPACER", "STRENGTH", and "TEMPERING".

The option code "ZGL" is used to choose a custom glass configuration in order entry. Nested beneath this option code are separate questions for each sash of this particular window. This custom option will allow the user to select different IG units for each sash in a window. See image 2 below:

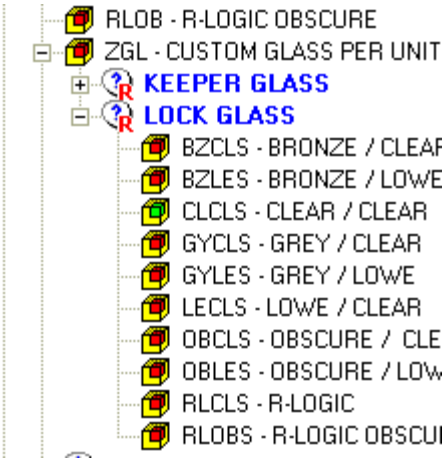


Image 2

As can be seen, the sash specific questions are "KEEPER GLASS" and "LOCK GLASS". The option codes beneath these questions are very similar to those found in the "IG" question. The reason they are not the same is due to pricing. The sash specific option codes (option codes with "S" appended to the end) will be priced at half the price that the regular glass option codes are priced.

Keep in mind that option codes can only be 6 characters in length.

Bill of Materials Structure:

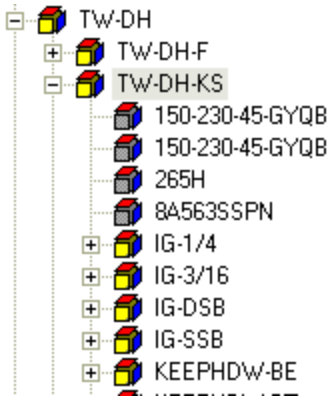


Image 3

Image 3 shows where the glass parts will be located. There are four glass subparts in this bill of materials, one for each available strength. Nested beneath these glass subparts is a part for each IG unit available in that strength. Each glass subpart is structured so that it can be used in any sash in any window type. This is so that these subparts can easily be reused. All attributes on the glass subparts are global.

Attributes:

H = *Height*

```
retval = Parent.Attributes("IGH").Value
```

I = *Include*

```
Include = attributes("IGS").value = "DSB"
```

IGS = *Insulated Glass Strength*

```
If OptionExists("DSB") Then
      retval = "DSB"
    ElseIf OptionExists("3/16") Then
      retval = "3/16"
    ElseIf OptionExists("1/4") Then
```

```
      retval = "1/4"
    ElseIf OptionExists("NSO") Then
      retval = OptionValue("NSO")
    End If
```

IGT = Insulated Glass Type

```
Dim StrLen As Integer
    Dim GroupText As String

    GroupText = parent.attributes("GLGT").value 'Ex: (GlassGroupText=actual text
to IG question)

    If OptionExists("ZGL") Then
      retval = GroupCode(GroupText) 'Ex:=Answer to IG question
      StrLen = Len(retval)
      retval = Left(retval,StrLen-1) 'strips the S (ash) off the end
    Else
      retval = GroupCode("IG") 'Answer to IG question (without S)
    End If
```

W = Width

```
retval = Parent.Attributes("IGW").Value
```

Grid Configuration

WARNING: This configuration is not for users ordering grids from an IG supplier! Grid configuration will be set up differently for WinIG and similar interfaces.

This section assists in the creation of FeneVision bill of materials cut logics for grids. This is a guideline and requires modification to best suit each individual manufacturers needs.

Options Structure:

Option codes may only be 6 characters long

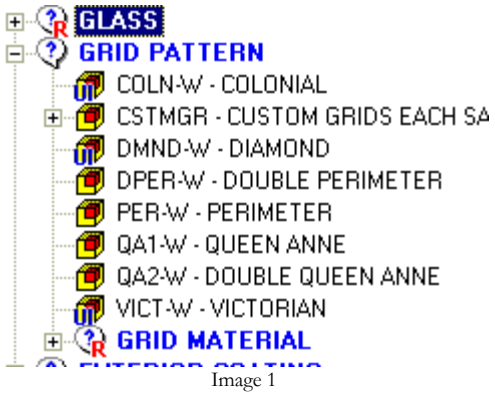


Image 1

Image 1 is an example of a commonly used options structure. Each grid pattern available is listed using a smart code that describes the pattern type.

Additional question regarding grids are found nested within the "GRID PATTERN" question. This is done so that if the order entry user were to pick "<none>" for the GRID PATTERN question, all the nested questions would be skipped. An example of a nested question, "GRID MATERIAL" is shown above. Other additional questions such as "DIFFERENT GRID COLOR" may also be added if available in order entry.

The option code "CSTMGR" is used to choose a custom gird configuration in order entry. Nested beneath this option code are separate questions for each sash of this particular window. This custom option will allow the user to select different grid patterns for each sash in a window. See image 2.

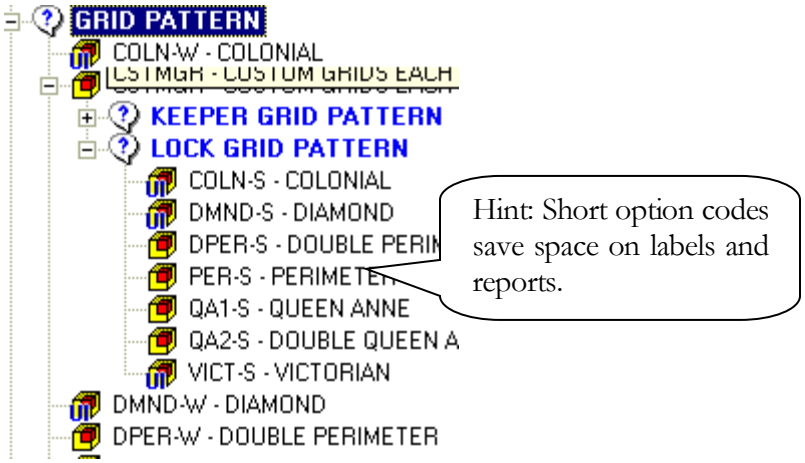


Image 2

As can be seen, the sash specific questions are "KEEPER GRID PATTERN" and "LOCK GRID PATTERN". The option codes beneath these questions are very similar to those found in the "GRID PATTERN" question. The reason they are not the same is due to pricing. The sash specific option codes (option codes with "S" appended to the end) will be priced at half the price that the regular grid option codes are priced.

Bill of Materials Structure:

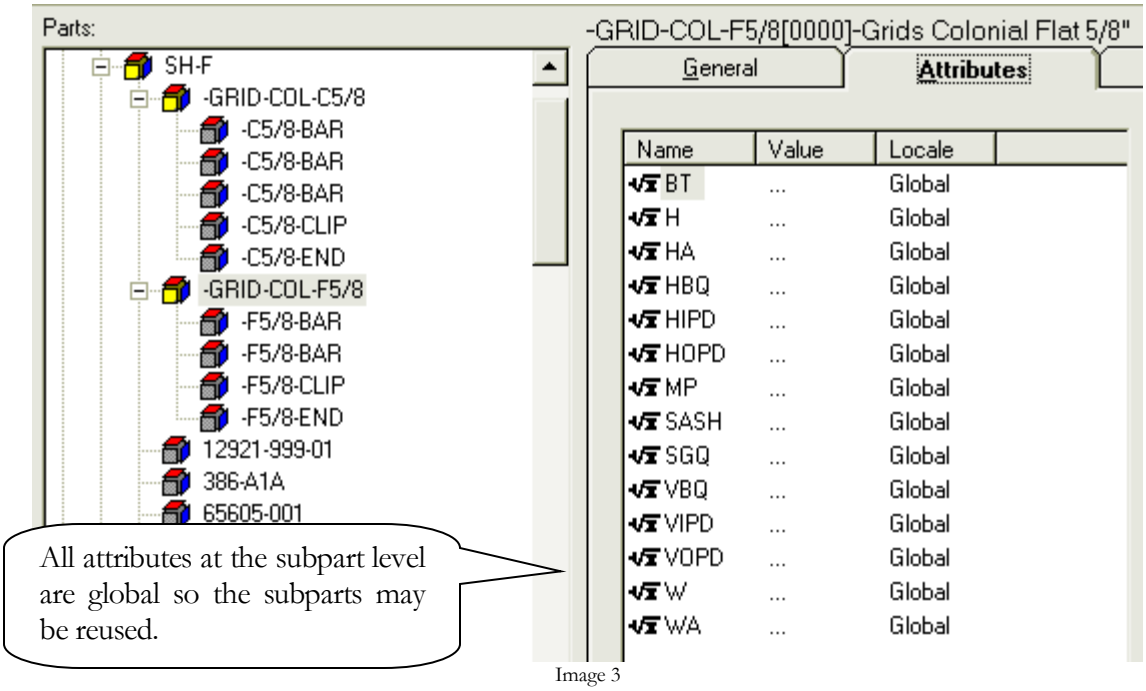


Image 3

Image 3 shows where the grid parts will be located. There will be a grid subpart for each pattern and each material combined. Two colonial subparts are shown in Image 3, one with flat material and one with sculptured material. Each grid subpart is structured so that it can be used in any sash and in any window type. This is so that these subparts can easily be reused.

Nested beneath these grid subparts are all the parts necessary in manufacturing the grids. The grid material is listed multiple times as can be seen in Image 3. This is due to the fact that there are multiple lengths of grid parts that need to be cut. For the Flat 5/8 grids there are two grid material parts listed (-F5/8-BAR). One part will return the height cuts, and one part returns the width cuts. For the contour 5/8 grids there are three parts listed (-C5/8-BAR). One part will return the height cuts, one part will return the inside width cuts, and one part will return the outside width cuts.

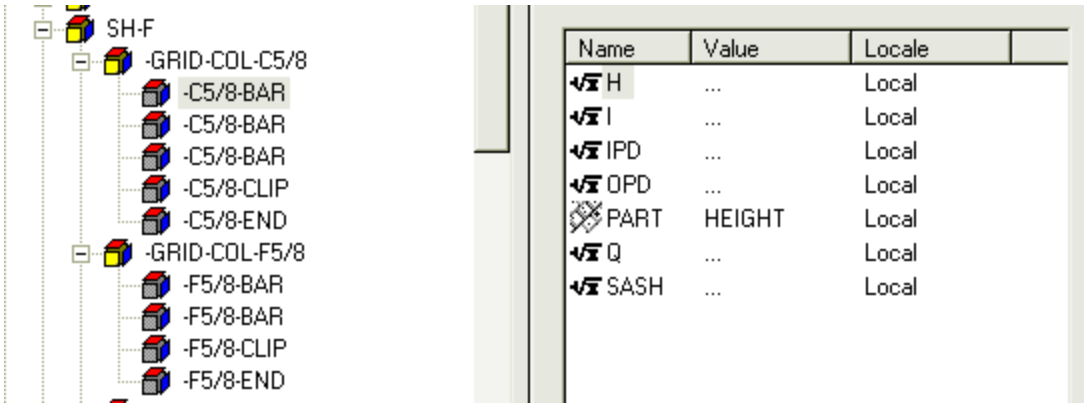


Image 4

Image 4 shows attributes created for one of the grid material parts. The IPD, OPD, PART, and SASH attributes will be shown on the grid report created. IPD and OPD will give the inner and outer punch

distances for inserting the internal clips (cross clips). PART will display what component of the grid is being cut. SASH will return what sash this grid material belongs in.

Attributes & Scripts Used:

Subpart Attributes & Scripts:

BT = Bar Thickness

```
'BAR THICKNESS
'Returns thickness of sculptured grid component used in calculations

If OptionExists("SCP5/8") Then
  retval = 0.34375  '11/32
ElseIf OptionExists("SCP1") Then
  retval = 0.5      '1/2
Else
  retval = "FLAT"
End If
```

H = Height

```
'GRID HEIGHT
'Returns Height of grid from spacer to spacer

retval = Parent.Attributes("IGH").Value - 0.75
```

HA = Height Adjust (returns glass size difference from parent part)

```
retval = Parent.Attributes("HA").Value
```

HBQ = Horizontal Bar Quantity

```
'HORIZONTAL BAR QUANTITY
'Calculates number of horizontal bars based on MP attribute

Dim MP As String
Dim POS1 As Integer

MP = Attributes("MP").Value     'MP = 2Vx2HC  <--example MP
If InStr(1,MP,"HC") Then        'Checks for existance of "HC"
  POS1 = InStr(1,MP,"HC") - 1   'Looks for position of "HC"
  retval = CInt(Mid(MP,POS1,1)) 'Extracts # from String
End If
```

HIPD = Horizontal Inside Punch Distance

```
'HORIZONTAL INSIDE PUNCH DISTANCE
'Calculates distance between horizontal inside punches (punches on horizontal
bars)

Dim VBQ,WA,W As Single
```

```
    VBQ = Attributes("VBQ").Value
    WA = Attributes("WA").Value
    W = Attributes("W").Value

    retval = (W + WA) / (VBQ + 1)
    If VBQ = 0 Then retval = 0
```

HOPD = Horizontal Outside Punch Distance

```
    'HORIZONTAL OUTSIDE PUNCH DISTANCE
    'Calculates distance for horizontal outside punches (punches on horizontal
bars)

    Dim VBQ,WA,W As Single

    VBQ = Attributes("VBQ").Value
    WA = Attributes("WA").Value
    W = Attributes("W").Value

    retval = ((W + WA) / (VBQ + 1)) - (WA / 2)
    If VBQ = 0 Then retval = 0
```

MP = Muntin (Grid) Pattern (will be specific based on options structure used)

```
    'MUNTIN (GRID) Pattern
    'returns grid pattern entered in order entry

    Dim SashGridQuestion As String
    SashGridQuestion = Attributes("SGQ").Value

    If OptionExists("CSTMGR") Then
      If OptionExists("COLN-S",SashGridQuestion) Then
            retval = OptionValue("COLN-S",SashGridQuestion)
          ElseIf optionexists("DMND-S",SashGridQuestion) Then
            retval = OptionValue("DMND-S",SashGridQuestion)
          ElseIf optionexists("VICT-S",SashGridQuestion) Then
            retval = OptionValue("VICT-S",SashGridQuestion)
          ElseIf optionexists("QA1-S",SashGridQuestion) Then
            retval = "QA1-S"
          ElseIf optionexists("QA2-S",SashGridQuestion) Then
            retval = "QA2-S"
          ElseIf optionexists("PER-S",SashGridQuestion) Then
            retval = "PER-S"
          ElseIf optionexists("DPER-S",SashGridQuestion) Then
            retval = "DPER-S"
          End If
    Else
          If optionexists("COLN-W") Then
            retval = OptionValue("COLN-W")
          ElseIf optionexists("DMND-W") Then
            retval = OptionValue("DMND-W")
          ElseIf optionexists("VICT-W") Then
            retval = OptionValue("VICT-W")
          ElseIf optionexists("QA1-W") Then
            retval = "QA1-W"
```

```
        ElseIf optionexists("QA2-W") Then
          retval = "QA2-W"
        ElseIf optionexists("PER-W") Then
          retval = "PER-W"
        ElseIf optionexists("DPER-W") Then
          retval = "DPER-W"
        End If
    End If
```

SASH = Grids Sash Location

```
    retval = Parent.Attributes("SASH").Value
```

SGQ = Sash Grid Question (returns sash specific option group text from parent)

```
    retval = Parent.Attributes("SGQ").Value
```

VBQ = Vertical Bar Quantity

```
    'VERTICAL BAR QUANTITY
    'Calculates number of vertical bars based on MP attribute

    Dim MP As String
    Dim POS1 As Integer

    MP = ATTRIBUTES("MP").VALUE     'MP = 2Vx2HC  <--example MP
    If InStr(1,MP,"HC") Then        'Checks for existance of "V"
      POS1 = InStr(1,MP,"V") - 1    'Looks for position of "V"
      retval = CInt(Mid(MP,POS1,1)) 'Extracts # from String
    End If
```

VIPD = Vertical Inside Punch Distance

```
    'VERTICAL INSIDE PUNCH DISTANCE
    'Calculates distance between vertical inside punches (punches on vertical
bars)

    Dim HBQ,HA,H As Single

    HBQ = Attributes("HBQ").Value
    HA = Attributes("HA").Value
    H = Attributes("H").Value

    retval = (H + HA) / (HBQ + 1)
    If VBQ = 0 Then retval = 0
```

VOPD = Vertical Outside Punch Distance

```
    'VERTICTAL OUTSIDE PUNCH DISTANCE
    'Calculates distance for vertical outside punches (punches on vertical bars)

    Dim HBQ,HA,H As Single

    HBQ = Attributes("HBQ").Value
    HA = Attributes("HA").Value
    H = Attributes("H").Value
```

```
retval = ((H + HA) / (HBQ + 1)) - (HA / 2)
If VBQ = 0 Then retval = 0
```

W = Width

```
'GRID WIDTH
'Returns width of grid from spacer to spacer

retval = Parent.Attributes("IGW").Value - 0.75
```

WA = Width Adjust (returns glass size difference from parent part)

```
retval = Parent.Attributes("WA").Value
```

Grid Material Attributes & Scripts:

H = Height (used in flat & contour grids)

```
retval = Parent.Attributes("H").Value
```

I = Include (for height cut)

```
If Parent.Attributes("VBQ").Value = 0 Then
  Include = False
Else
  Include = True
End If
```

I = Include (for outside width cut)

```
If parent.attributes("HBQ").value = 0 Then
  Include = False
Else
  Include = True
End If
```

I = Include (for inside width cut)

```
If Parent.Attributes("VBQ").Value <= 1 Then
  Include = False
ElseIf parent.attributes("HBQ").value = 0 Then
  Include = False
Else
  Include = True
End If
```

IPD = Inside Punch Distance

```
retval = Parent.Attributes("VIPD").Value
```

OPD = Outside Punch Distance

```
retval = Parent.Attributes("VOPD").Value
```

PART = *Grid Part Type*

String with text = "HEIGHT"

Q = *Quantity (for height cut)*

```
retval = Parent.Attributes("VBQ").Value
```

Q = *Quantity (for outside width)*

```
If parent.attributes("VBQ").value = 0 Then
  retval = Parent.Attributes("HBQ").Value
Else
  retval = Parent.Attributes("HBQ").Value * 2
End If
```

Q = *Quantity (for inside width)*

```
retval = Parent.Attributes("HBQ").Value * (Parent.Attributes("VBQ").Value -
1)
```

SASH = *Grids Sash Location*

```
retval = Parent.Attributes("SASH").Value
```

W = *Width (outside: used in contour grids)*

```
'Outside Widths

If Parent.Attributes("VBQ").Value = 0 Then
  retval = Parent.Attributes("W").Value
Else
  retval = Parent.Attributes("HOPD").Value -
  (Parent.Attributes("BT").Value / 2)
End If
```

W = *Width (inside: used in contour grids)*

```
'Inside Widths

retval = Parent.Attributes("HIPD").Value -
(Parent.Attributes("BT").Value * (Parent.Attributes("VBQ").Value - 1))
```

Grid Report Setup:

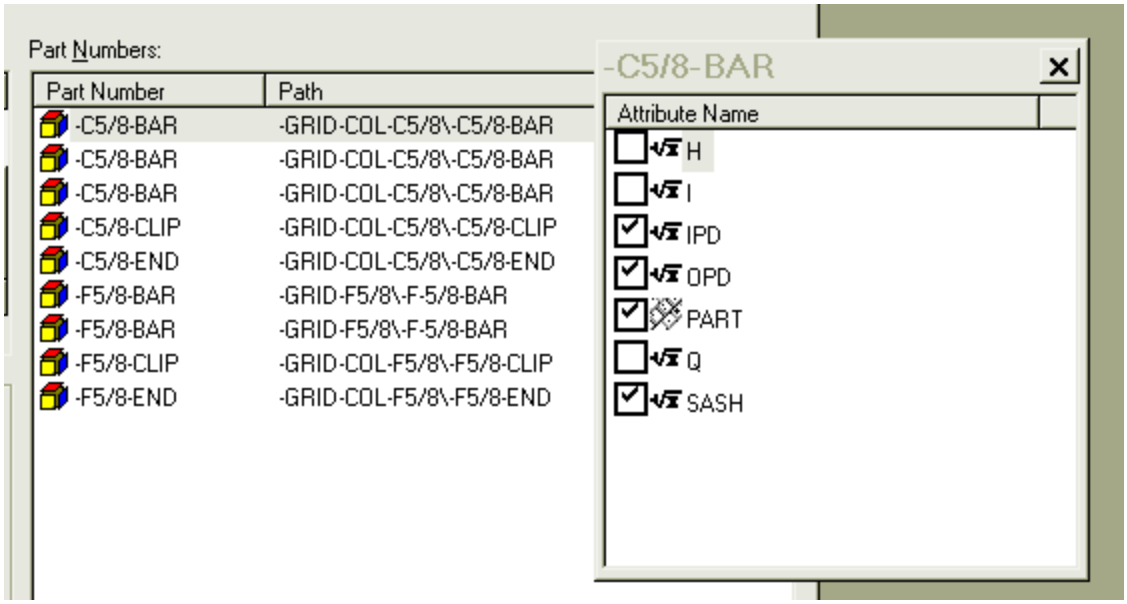


Image 5

Image 5 shows the report setup for a general grid report. Each of the inventoried grid parts are dragged into the report setup screen from within the subpart level. This is noted by the path displayed in the reports interface. Since the part is dragged from within a specific subpart the local attributes are available to be displayed. IPD, OPD, PART, and SASH are selected because they are necessary information for building grids. Some parts such as the inside and outside widths (on contour grids) do not have punches. In that case they will not be displayed on the reports.

Balance Configuration

The sash weigh is used to determine what balances are included in the bill of materials for a particular window. This sash weight is calculated using attributes in the FeneVision bill of materials, and the balance parts are included based on the weight calculated in these attributes.

The sash weight should be calculated using attributes at the frame part level. This way an attribute can be shown on the frame label that indicates what balance is to be used in a particular frame. This also serves as a single place all the frame sub-parts can reference in their include statements.

Formula for Sash Weight:

The Sash Weight is obtained by calculating the weights of each piece of material that compromises the sash. The following equation displays the basic calculation for determining sash weight.

Sash Weight = Vertical Vinyl Weight + Horizontal Vinyl Weight + Glass Weight + Hardware Weight – Friction

Or

SW = **VVW** + **HVW** + **GW** + **HW** - **F**

Each attribute described in more detail:

Vertical Vinyl Weight:

VVW = Weight of each Stile + Weight of each vertical Glazing Bead

= ((Stile Weight Factor * Sash Height) * 2) + ((Glazing Bead Weight Factor * (Sash Height – GBCB)) * 2)

GBCB = Glazing Bead Cut Back

The Weight factors are Generally given in lbs/lft. The dimensions in Production Control are always in inches so, be sure to perform the proper conversions. The above formula, with Weight Factors of lb/lft would read:

VVW = (((SWF/12) * SH) * 2) + (((GBWF/12) * (SH – GBCB)) * 2)

Horizontal Vinyl Weight:

HVW = Weight of each Rail + Weight of each horizontal Glazing Bead

= ((Rail Weight Factor * Sash Width) * 2) + ((Glazing Bead Weight Factor * (Sash Width – GBCB)) * 2)

GBCB = Glazing Bead Cut Back

The Weight factors are Generally given in lbs/lft. The dimensions in Production Control are always in inches so, be sure to perform the proper conversions. The above formula, with Weight Factors of lb/lft would read:

HVW = (((SWF/12) * SW) * 2) + (((GBWF/12) * (SW – GBCB)) * 2)

Glass Weight:

IGP = IG Perimeter (inches)

= (((2 * Glass Height) + (2 * Glass Width)) / 12)

SPWF = Spacer/Sealant Weight Factor (lbs/lft)

GWF = Glass Weight Factor (lbs/sqft)

GW = (((Glass Height * Glass Width) / 144 * GWF) * 2)+ ((IGP / 12) * SPWF)

Hardware Weight:

HW = Total weight of Locks, Keepers, vent locks, etc.

Note: Adjust Hardware Weight for single and double locks

Friction (optional):

Friction = Sash Height * Coefficient of Friction

* The coefficient of friction will very based on how tight the sashes fit in the Frame.

Common weight factors:

GWF:

```
'TS
If  (using 3/16" glass) then
      Retval = 2.29 lb/sqft
'DS
ElseIf  (using 1/8" glass) then
      Retval = 1.53 lbs/sqft
'SS
Else
      Retval = 1.15 lbs/sqft
End if
```

Rail Weight Factor:

.243 lbs/lft

Stile Weight Factor:

.263 lbs/lft

Coefficient of Friction:

.05 lbs

Balance Attribute:

The balance attribute is commonly named BAL. This attribute uses the sash weight to calculate the appropriate balance size needed in a window. It will return a text value that will be shown on frame labels as the balance needed. Balance sub-parts will also use this text value in their include statements. Below is an example of a BAL attribute:

```
Dim SashWeight As Single

SashWeight = Attributes("BSW").Value

   If SashWeight < 9 Then

         retval = "B"

   ElseIf ((SashWeight  > 9) And (SashWeight <= 11)) Then

         retval = "C"

   ElseIf ((SashWeight  > 11) And (SashWeight <= 13)) Then

         retval = "D"

   ElseIf ((SashWeight  > 13) And (SashWeight <= 14)) Then

         retval = "E"

   ElseIf ((SashWeight  > 14) And (SashWeight <= 16)) Then

         retval = "F"

   ElseIf ((SashWeight  > 16) And (SashWeight <= 20)) Then

         retval = "BCL"

   End If
```

Including Sub-Parts:

Balance sub-parts such as balance coils or balance shoes will need to reference the BAL attribute in their include statements. Below is a sample include statement:

```
Dim BAL As Single

BAL = Parent.Attributes("BAL").Value

If (BAL = "B") Then

    Include = True

Else

    Include = False

End If
```