

BOM 明细表的制作

by 无维网 IceFai (黄光辉)

在 Proe 中 BOM (Bill Of Material) 表的制作关键技术由三个部分组成: Table (表), Repeat Region (重复区域) 和 BOM Balloons (球标)。下面我们就按这三个部分进行详细的讲解。

Table (表)

指令位置: Table (表) → Insert(插入) → Table(表)

对于表的操作又可以分为两大类: 表格式操作和表格内容操作。表格式操作和一般的表的格式处理类似, 你可以对表: 创建, 增加或减少行/列, 分成多部分, 控制表格线的显示与否, 调整行/列/单元格的高度/宽度。表格内容操作包括文本输入, 报表参数输入, 文本换行等。

表 (Table) 的创建

对于表的创建过程来说, 有三个选项来确定它的创建方式: 行的展开方式, 列的展开方式和宽度度量方式。

Descending(下降)

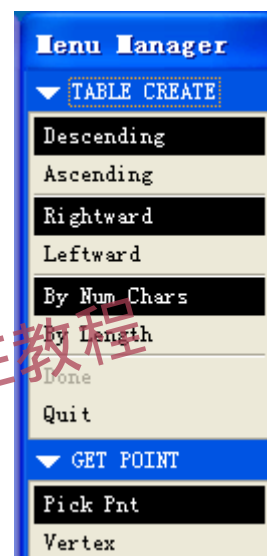
Ascending (上升)

Rightward (右展)

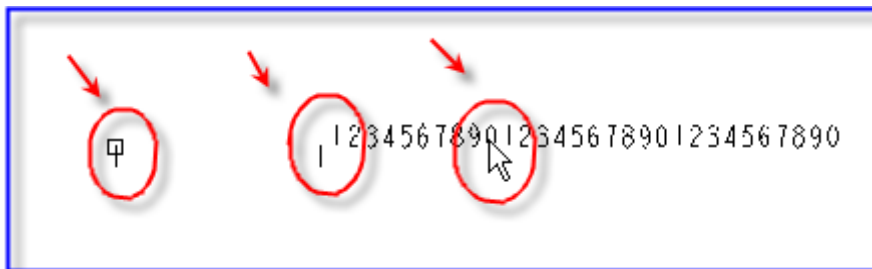
Leftward (左展)

By Num Chars (按字符数)

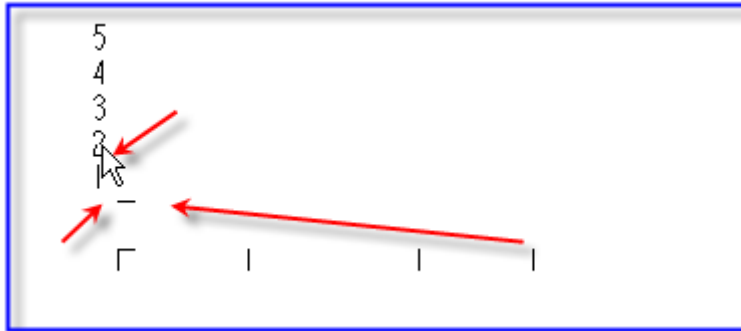
By Length (按单位长度)



要创建表, 首先点指令, 然后根据点的选取方法来在图形区中选确点作为表的起始点。如果你是用 By Num Chars (按字符数) 选项的话, 那么就会在你的选取点旁边出现一行数字。如下图:



在数字中点取一个就表明你要创建对应列宽的列, 如点击第一个 0 表明创建列宽为 10 个字符的列, 点第二个 3 的话就是列宽为 13, 以此类推。当所有列都创建完毕时可以按中键以结束列的创建, 系统自动转为行的创建, 并且在起点的上下方向出现另一列数字, 相应的在数字上点击就是创建对应行高的表格行, 同样可以重复创建直到达到需要的表格行数。

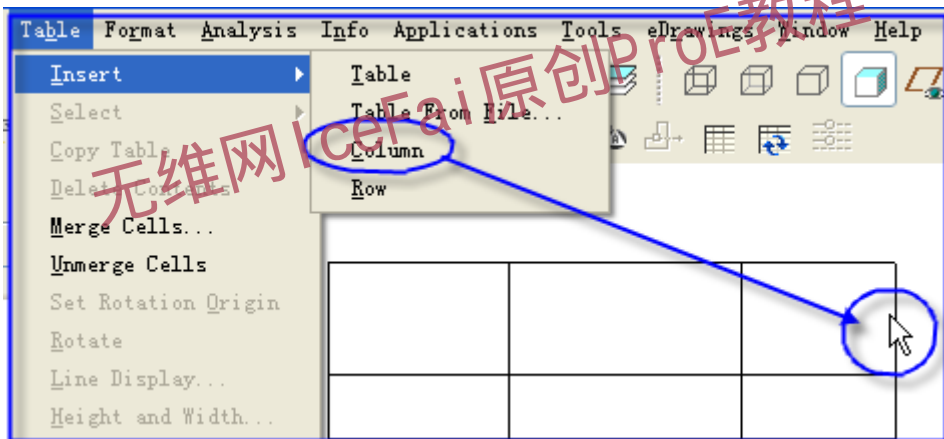


如果你是以 by Length（按单位长度）选项来创建的话，当点取点后系统就会提示你输入表格的宽度（或高度），单位根据配置选项中设定的单位。

增加一行（Row）或列（Column）

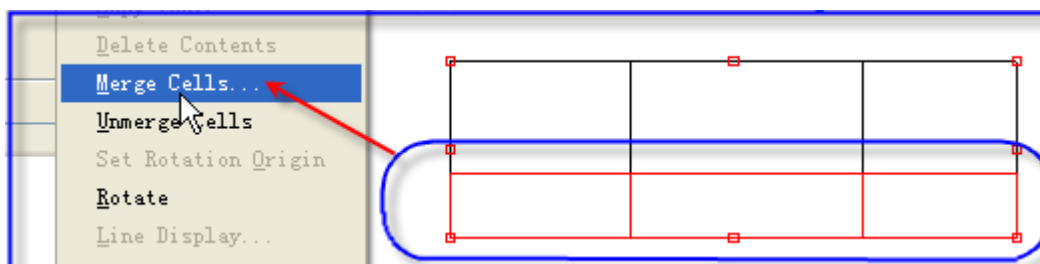
当你的表创建后之后，在使用过程中发现表的行数或列数不够或者太多的时候，你可以随时增加或删除一行（Row）或列（Column）。要删除一行或列很简单，只需要选择表中要删除的行或列，然后按 delete 键或鼠标右键菜单中选择 delete 便可。要注意的是，在表中，你选取的对象可能有四种：表格，行，列，单元格；要区分它们，你只需看选择的时候加亮的线条包括哪些就很容易知道自己当前选择的到底是什么了。当发现不是自己所需的可以按右键以选择下一项内容。

而要增加一行或列，选取菜单 Table(表)→insert(插入)→Row（或 Column）然后在已有的表格中要增加的位置的相邻边上点击一下就可



合并（Merge）和拆分（Unmerge）单元格

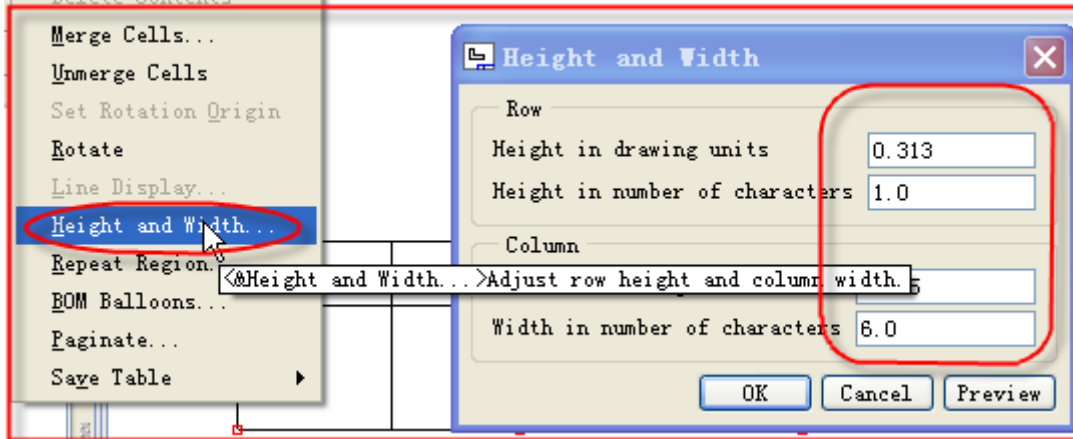
要合并单元格，首先按住 Ctrl 键来选择要合并的相邻表格，然后点菜单 Table→Merge Cells... 便可，表格合并后中间的线条自动消失，原来的多个单元格变成一个单元格。



要把合并得来的表格重新恢复原状（多个原始单元格）你可以用 Table 菜单下的 Unmerge Cells，同样只需在选择单元格的时候点菜单指令便可。

改变表格的行和列的宽（Width）/高度（Height）

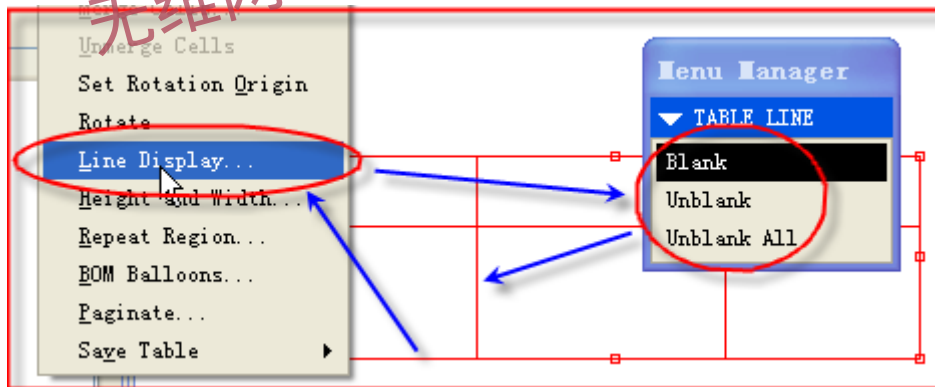
你可以随时修改表格的行/列的宽和高度。选中表格（或行，列，单元格），这样就会激活 Table 菜单下的 Height and Width...选项（要改变特定的行或列的宽和高度可以在右键菜单中选择），这样就会激活高度和宽度的修改对话框。如下图所示



在对话框中，你可以用单位来输入也可以用字符数目来输入，系统都会自动更新相应的单位数或字符数。

边框线的显示（Line Display）

你也可以控制表格中的任意一条边框线的显示与否。这个要用到 Table 菜单下的 Line Display...项，这个项只有在选中表格元素（可以是表格，列，行和单元格）的时候才会激活。点 Blank 然后就可以点击表格中的框线来隐藏起来，Unblank 自然是重新显示出来，Unblank All 是显示所有隐藏的框线。

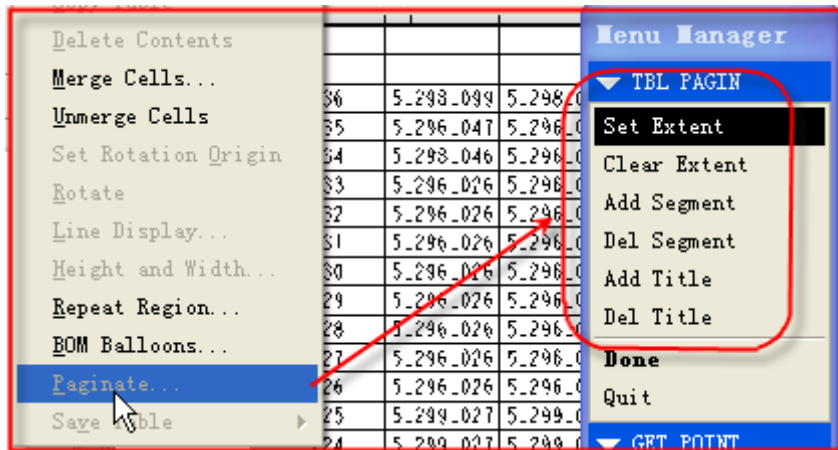


分页（Paginate）

当一个表太长的时候，或许你想把它分成几部分。这时候就要用到分页。分页功能只有对带重复区域（Repeat Region）的表格才会生效的。

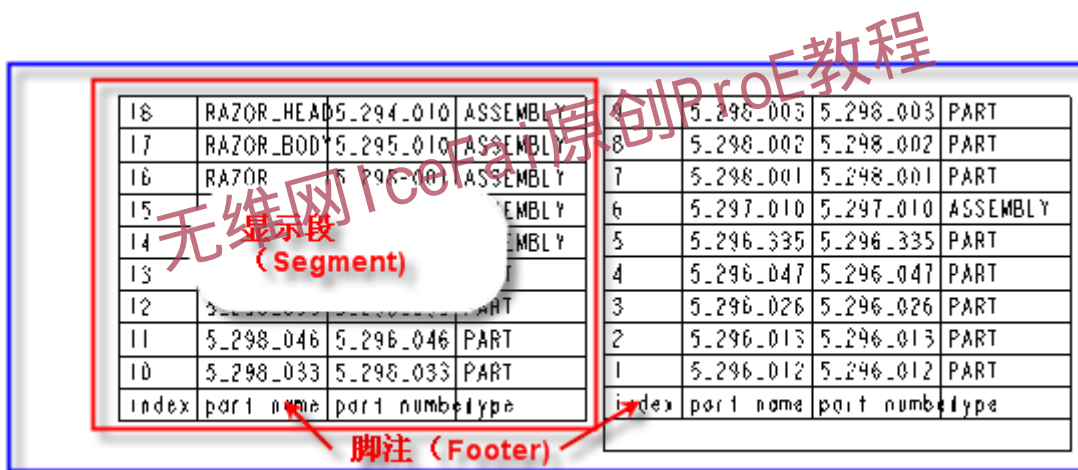
指令位置：Table(表)→Paginate...(分页)。只有在选中带重复区域表格的前提下，这个指令才会激活。

具体的操作如下：首先我们选中表，然后点 Table→Paginate...。系统弹出 TBL PAGIN 菜单，



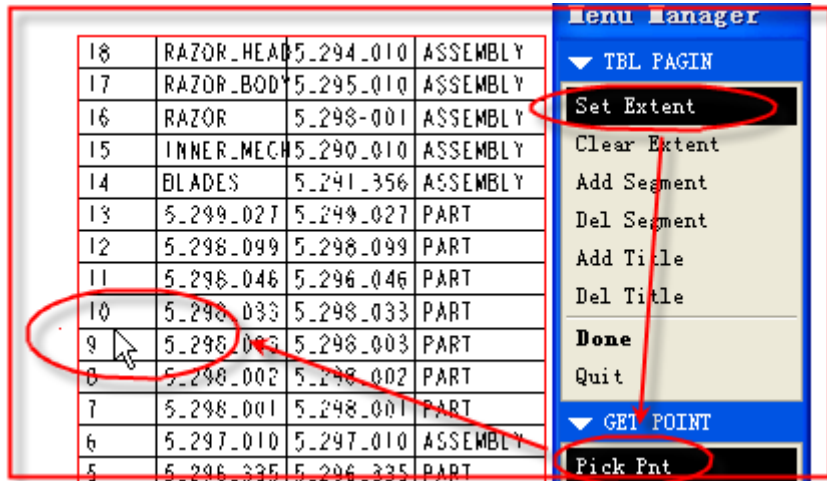
菜单项如下：

- ✧ Set Extent（设置范围）--设定要分页的范围，也就是指定开始分页的地方。
- ✧ Clear Extent（清除范围）--取消分页。
- ✧ Add Segment（增加显示段）---把分页范围内的表内容用新段显示出来
- ✧ Del Segment（删除显示段）
- ✧ Add Title（添加标题）--为重复区域添加标题，标题有两种：Header（页眉）和 Footer（脚注），每个显示段有
- ✧ Del Title（删除标题）



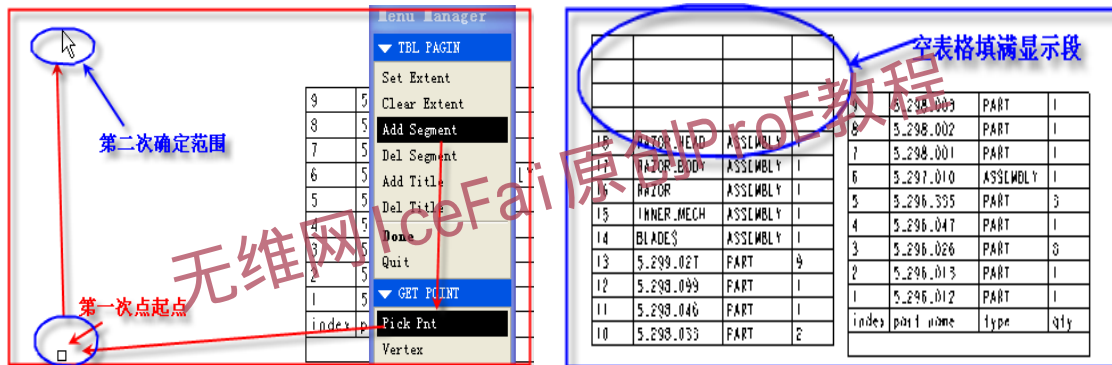
Set Extent(设置范围)

在这理的范围其实就是分页的起点，只需要在表格要进行分段的地方点击一下便可。表格中在点击位置后面的所有表格就构成了范围。一个范围可以用多个显示段来显示出来。表格设置范围后，表格就会在范围起点处断开。范围内的表格“消失”不见。



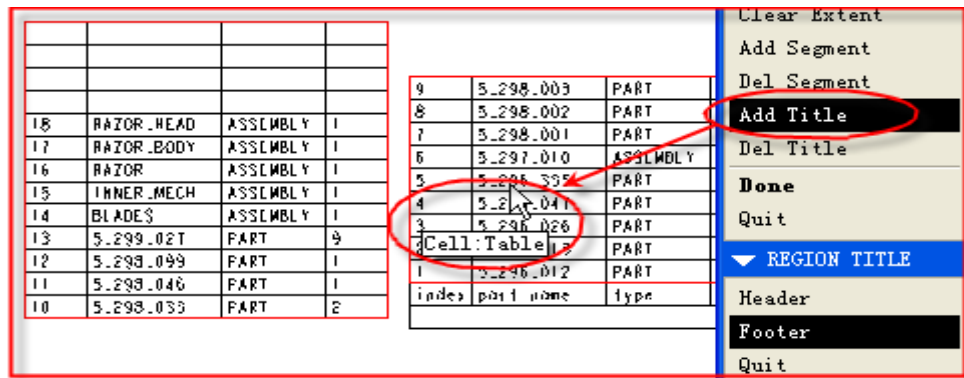
Add Segment (增加显示段)

当已经定义好范围后，你可以用 Add Segment (增加显示段) 来把范围内的表格在新的地方显示出来。显示段的长度有用户两次点击的位置来确定，不足的地方填以空表格。单一个显示段不足与全部显示范围内的表格时，你也可以添加多个显示段，系统会自动根据索引号递增而不会发生显示段重复的现象。



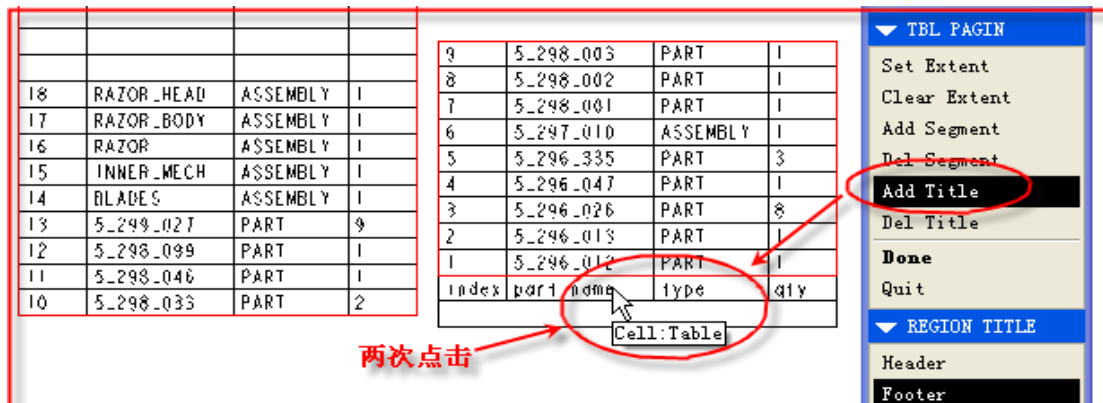
Add Title (增加标题)

每一个重复区域都可以有一个脚注 (Footer) 或页眉 (Header)。不过这个重复区域如果有多个显示段 (Segment) 的话，那么每个 Segment 都有一个标题。要替重复区域添加标题，首先点 Add Title，然后选择一个重复区域。如下图所示：

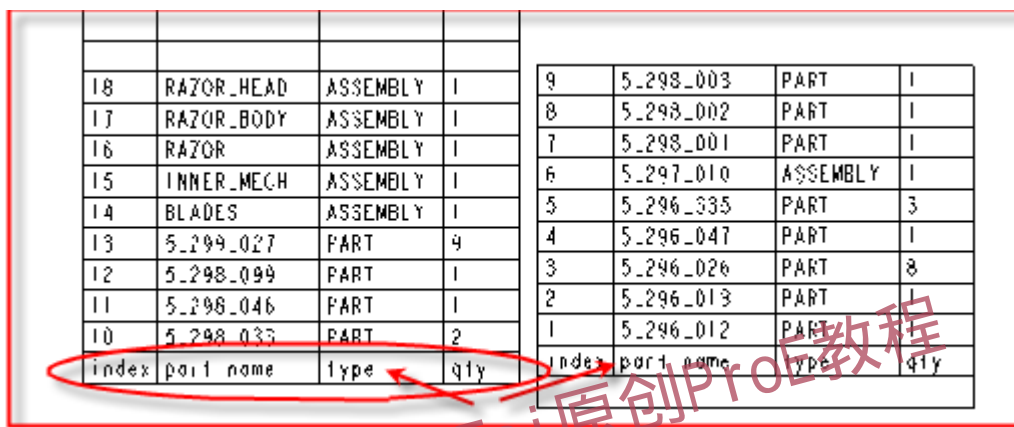


然后就是选择你要添加的标题类型 (Header 或 Footer)。接着选择要作为标题的表格行起点和终点，如果只想一行作为标题栏，那么两次点击都点该行便可。作为标题栏的行要和重复

区域在同一个表内但不能是重复区域内。

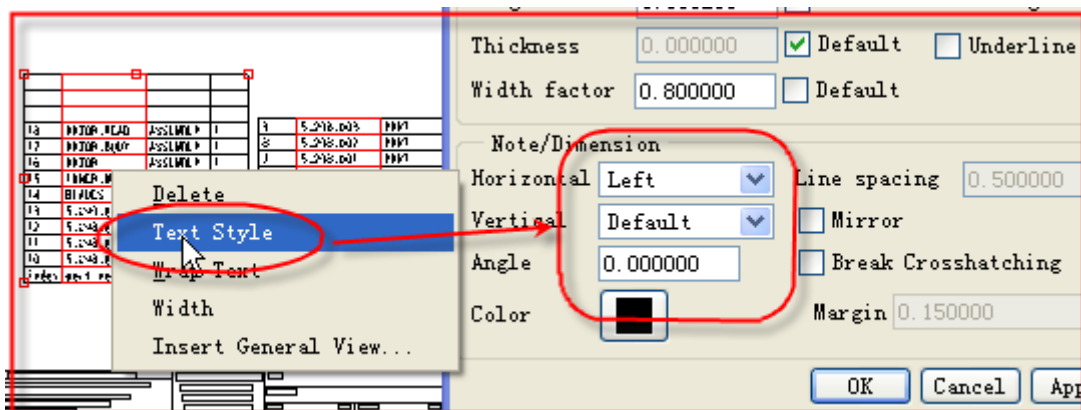


如下图显示，当标题行定义好后，另一个显示段内也会自动加上了标题行。



表格文本的对齐方式

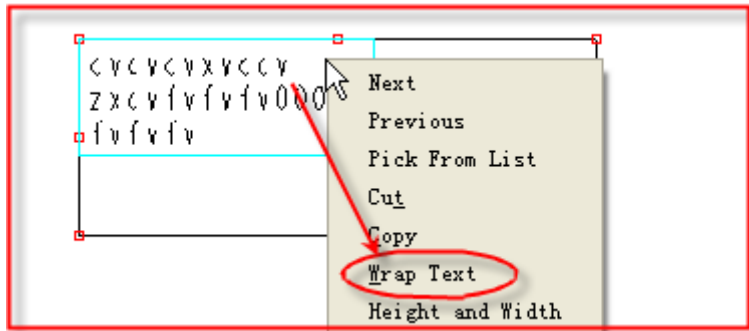
表格中的文本可以采用不同的对齐方式，要修改文本的对齐方式，只需在选中表格的时候用右键菜单中的 Text Style（文本样式）选项就可以打开 Text Style 对话框，在对话框中就可以选择表格文本在水平和竖直两个方向上的对齐方式。如下所示：



同时在这个对话框中也可以修改文本的字体，大小，线宽等。

表格文本的自动换行（Wrap Text）

有时，表格的单元格宽度所限，容不下所有文本的时候，你可以用换行的方式来在单元格中以多行显示表格内容文本。



重复区域 (Repeat Region)

所谓的重复区域 (Repeat Region)，就是表中用户指定的变量填充的部分，这部分会根据相关模型所含的数据量的大小相应的进行展开或收缩以显示所有符合条件的数据。重复区域的信息是由基于文本的报表符号来决定的，它们以文本的形式填充到重复区域内的表格中。

动态展开收缩是重复区域的最大特点，例如重复区域相关的装配有 20 个零件的话，在区域的一个表格内输入 `asm.mbr.name`，那么在表更新的时候会自动展开成 20 行以便为每个零件创建一个对应的单元格。

下面我们来看一个简单的例子：假设我们有一个表如下图所示，其中底下第一行就是重复区域，区域中的各个单元格的报表符号如图所示。

索引	零件名称	数量
<code>rpt.index</code>	<code>asm.mbr.name</code>	<code>rpt.qty</code>

那么当表更新的时候，它的展开应该类似下图所示

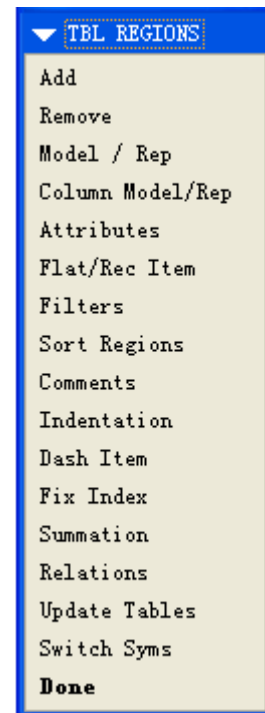
索引	零件名称	数量
1	COVER_FRONT	1
2	轴	1
3	衬套	2
4	轴承	4
5	COVER_BACK	1

正是因为重复区域的这个重要特性，所以非常适用于用来自动生成装配的 BOM 表。

指令位置：Table → Repeat Region... (重复区域)

在表中可以创建重复区域，当然也可以对重复区域进行各种处理，具体的菜单项及作用如下图所示：

- Add (添加) --创建一个重复区域
- Remove (移除) --删除一个重复区域
- Model/Rep (模型/简化表示) --设定区域相关的模型和表示
- Column Model/Rep—设定数量列对应的模型和表示
- Attributes (属性) --设定重复区域的各项属性
- Flat/Rec Item (平铺/递归 项目) --设定单个项目的平铺和递归
- Filters (过滤) --设定重复区域的过滤规则或单个项目
- Sort Regions(排序)—设定区域内容的排序方法
- Comments(注释)—设定重复区域内的注释列
- Indentation(缩进)—设定区域不同级别间的项目的缩进量
- Dash Item (省略项) ——符号值用“-“代替(对 rpt.index 和 qty)
- Fix Index (固定索引) --设定项目的索引号
- Summation(求和)—对特定项进行求和
- Relations (关系) —添加或修改报表关系
- Update Tables (更新表)
- Switch Syms (切换符号) --在符号和值之间进行切换。

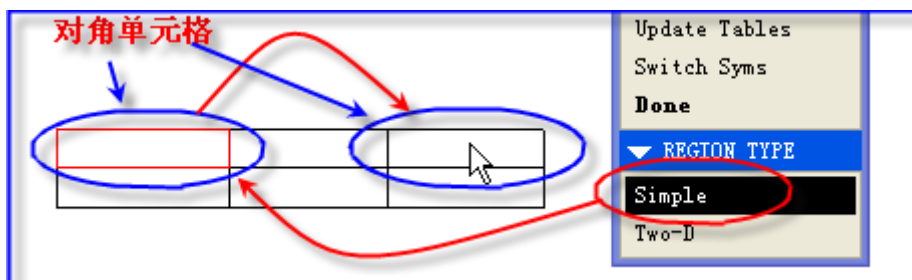


Add (添加) 区域

重复区域有两种，根据展开的方式来分类：Simple (简单) 和 Two-D(2D)。简单报表 (Simple) 只向表的上下方向展开 (单方向展开)，而 2D (Two-D) 重复区域则同时在上下和左右两个方向展开，一般用于族表零件。

简单重复区域 (Simple) 的创建

简单重复区域的创建比较简单，你只需要分别点击定义区域的两个对角单元格就行了。简单的 (Simple) 重复区域可以由多行构成，不过一般来说都是单行的。

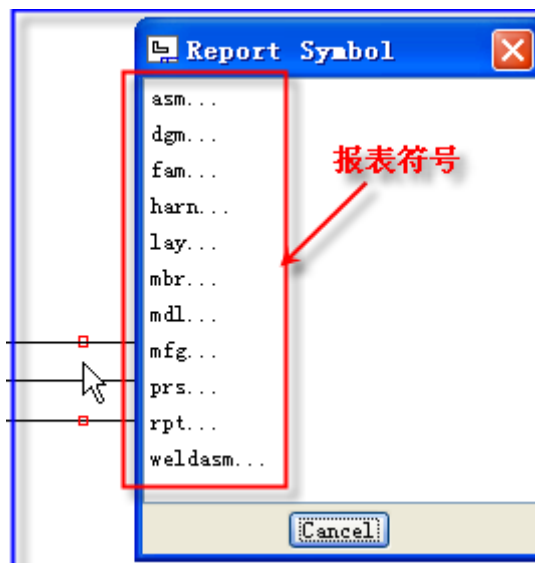


这样你的重复区域表格就创建好了，现在你在其中的一个表格中双击一下，就会看到系统并不是象一般的表格那样弹出一个 Text Style 的对话框而是一个报表符号 (Report Symbol) 的对话框。

对话框中的项目就是报表符号 (Report Symbol)，所谓的报表符号，其实就是一个参数，这个参数会自动根据对应的模型更新它的值，不同的模型有不同的对应值。

你看到的符号右边有...的就表明这个符号有子项可以展开。报表符号分为下面的几大类：

asm...---有关装配的信息
 dgm...---
 fam...---有关族表的信息
 harn...---有关电缆的参数信息
 lay...---有关 layout 的信息
 mbr...---有关单个元件的信息
 mdl...---有关单个模型的信息
 mfg...---有关 mfg 的信息
 prs...
 rpt...---有关重复区域的信息
 weldasm...有关焊接装配的信息



尽管报表符号有很多，但是常用的也不算太多，记住下面的这些足以应付你的一般工作：

asm.mbr.name	装配中的成员名称
asm.mbr.type	装配中的成员类型（Assembly 或 Part）
asm.mbr.(user defined)	装配中的成员的用户自定义参数
rpt.index	报表的索引号
rpt.qty	报表中的成员数量
rpt.level	报表中的成员所处的装配等级
rpt.rel.(user defined)	报表关系中的用户自定义参数
fam.inst.name	族表的实例名
fam.inst.param.name	族表实例的参数名
fam.inst.param.value	族表实例的参数值

如下图的重复区域，区域内表格单元的报表符号分别是：rpt.index, asm.mbr.name,asm.mbr.VENDOR,rpt.qty.其中的 VENDOR 就是一个自定义参数，用来表示供应商的。

rpt.index	asm.mbr.name	asm.mbr.VENDOR	rpt.qty
index	part name	vendor	qty

用户自定义参数

那么这样的重复区域展开后就会类似下面一样

13	5.299.027	STOCK	9
12	5.298.099	MODELWORKS	1
11	5.298.046	MODELWORKS	1
10	5.298.033	STOCK	2
9	5.298.003	IN HOUSE	1
8	5.298.002	INJECTION LABS	1
7	5.298.001	INJECTION LABS	1
6	5.297.010	VENDOR PURCHASE	1
5	5.296.335	CAST PRODUCTS	3
4	5.296.047	STOCK	1
3	5.296.026	STOCK	8
2	5.296.013	MOLDHAUS	1
1	5.296.012	MOLDHAUS	1
index	part name	vendor	qty

TWO-D (2D) 重复区域的创建

2D 重复区域因为有两个方向的展开，所以相对创建也复杂些。首先我们来看一个典型的 2D 展开重复区域：

047-WN	50.000	6.000				N	4.000
047-OH	50.000	5.000				Y	4.600
047-32	50.000	3.200				Y	4.000
047-MS	50.000	3.000	12.000	20.000	3.400	N	4.600
disto	grip_ten	head_dia	head_rad	wack_dia	slot		thread_dia

对于带族表的零件来说，我们要用一个表格来查看所有实例的尺寸信息。那么我们来考虑一下这个表的结构：首先对于每个实例来说，都和普通模型有一样的参数和不同的参数值，每个参数值都有对应的实例和参数名。对于创建者来说，如果逐个添加实例的参数话不但麻烦而且当参数发生改变时表就失去了意义，重复区域如果能自动检索族表中的所有参数并展开的话这个问题就解决了。因为实例的展开和实例参数的展开是互不相关的，换句话说便是 2 维的。明白 2D 重复区域的特点，再来看它的创建过程就容易明白多了。

首先创建一个 2X2 的表格。然后在 Table 菜单中选择 **Repeat Region(重复区域)>Add (添加) >Two-D** 以创建一个 2D 重复区域。当提示 "Locate corners of the Region (选择区域的角点)" 时首先选择 A 和 D。然后在提示 "Select a cell to set the upper border of the row & column subregions" 时选择单元格 D。这样就创建了一个向下和向右展开的 2D 表格。对于不同的表格展开规则，按照下面的选择方法来进行选择：

A	B
C	D

如果表格：

- 降序(Descending)和右向，选择单元格 A 和 D

- 降序(Descending)和左向, 选择 B 和 C
- 升序(Ascending)和左向, 选择 D 和 A
- 升序(Ascending)和右向, 选择 C 和 B

当提示"Select a cell to set the upper border of the row and column subregions"时则按下面规则来选择:

如果表格是:

- 降序(Descending)和右向, 选择 D
- 降序(Descending)和左向, 选择 C
- 升序(Ascending)和左向, 选择 A
- 升序(Ascending)和右向, 选择 B

或许这样看你觉得晕, 其实规则很简单, 你只需要记住: **先点击要两方向展开的单元格的对角格一次, 然后再点击要两方向展开的单元格两次便可。**

如本例中, 显然 D 是要两个方向展开的单元格, 所以首先要点 D 的对角也就是 A, 然后在点击两次 D 便可。

单元格中报表符号的输入也按照展开的方式来输入: 首先在两个方向展开的单元格内输入对应的报表符号, 如: fam.inst.param.value (族表实例的参数值)。然后输入它在行上和列上的对应格的报表符号, 如果行: fam.inst.name, 列: fam.inst.param.name 便可。

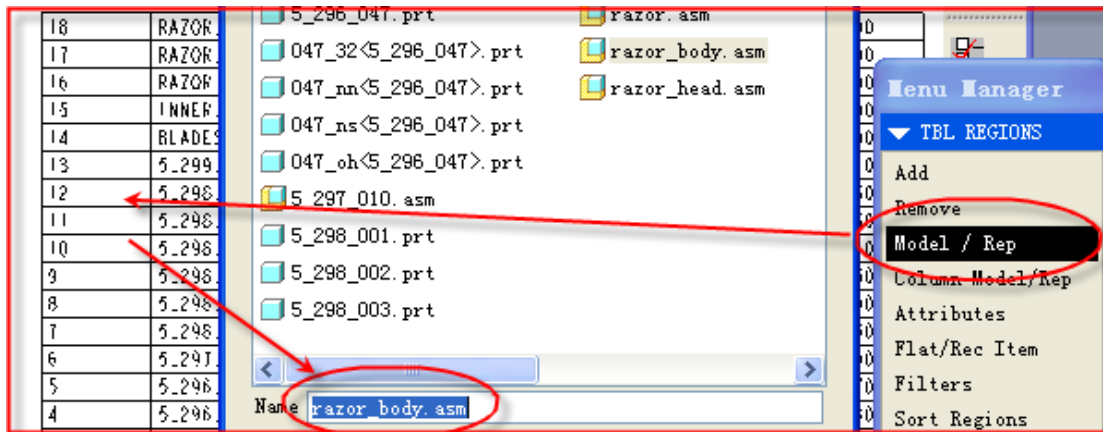
	fam.inst.param.name
fam.inst.name	fam.inst.param.value

上面的重复区域展开后应该类似如下所示, 注意实例名和参数值的对应:

	grip_len	head_dia	head_rad	neck_dia	SLOT	thread_dia
047_NS	3.000	12.000	20.000	3.400	N	4.600
047_32	3.200	12.000	20.000	3.400	Y	4.000
047_OH	5.000	15.000	25.000	3.400	Y	4.600
047_NN	8.000	12.000	20.000	4.000	N	4.000

Model/Rep(模型/简化表示)

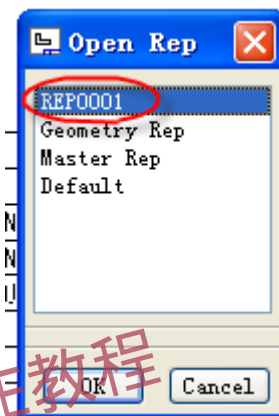
在重复区域中, 你可以设定重复区域所关联的模型和简化表示。首先选择 Model/Rep, 然后点击一个重复区域就会弹出一个对话框, 在对话框中你可以选择你想关联的模型。



如果所关联的模型中有简化表示的话，在选定文件后就会弹出选择简化表示的对话框。选择你想采用的简化表示。OK后完成，重复区域就会自动采用选择的模型和简化表示来更新。

简化表示中所排除的成员将不会在重复区域中显示或累加数目。

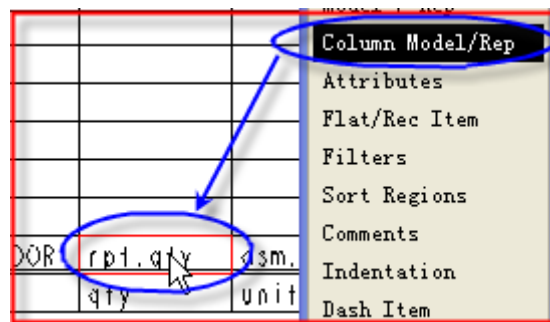
通常用于同一个表内有多个重复区域分别对应不同的子装配的情况。



Column Model/Rep (列模型/简化表示)

使用 Column Model/Rep 选项可以指定使用报表符号 rpt.qty 的列的关联模型和简化表示，当指定模型或简化表示后，那么对应的成员的数目将不再按重复区域的装配中的成员数目计算，而是按照指定的模型和简化表示中所包含的成员数目计算。

根据它的作用不难知道它只能对那些使用 rpt.qty 作为报表符号的重复区域中的列才能使用。



属性 (Attribute)

重复区域的属性只要有下面的几组，每一组中只能选一项作为属性之一：

◇ 重复性

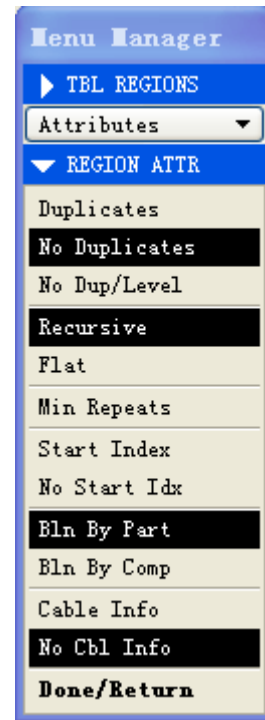
Duplicates: 重复，重复的成员当作独立的成员处理

No Duplicates: 无重复，所有重复的成员当作一个成员的成员累加计数

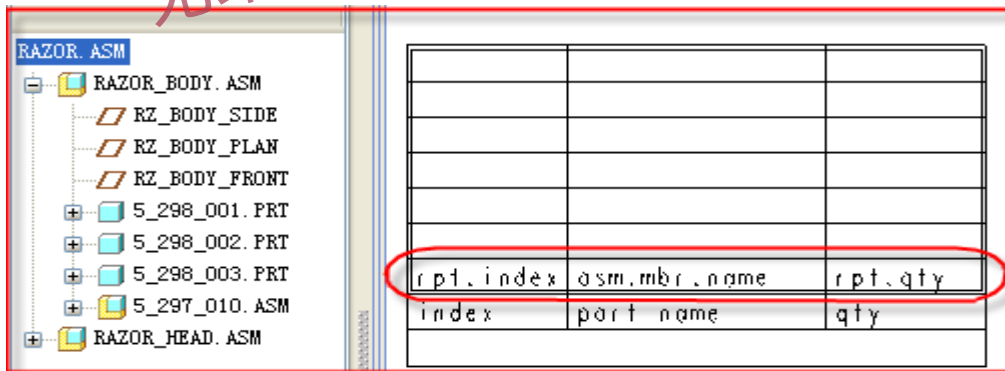
No Dup/Level: 无重复/级别，只有同一装配等级的重复成员当作一个成员并累加计数。

◇ 展开方式

- Recursive: 递归, 展开装配内的所有等级子装配
- Flat: 平铺, 只展开模型的下一级成员
- ✧ 最小重复数
 - Min Repeats: 最小重复数, 设定所选的重复区域的可能最小重复数
- ✧ 索引方法
 - Start Index: 起始索引, 把当前重复区域的索引起始号设为紧接另一个重复区域的终止索引号。
 - No Start Idx: 无起始索引号, 当前重复区域的索引号从 1 开始
- ✧ 丢失零件球标的附着方式
 - Bln By Part: 丢失的零件的球标附着到丢失零件上
 - Bln By Comp: 丢失的零件的球标附着到所在的成员上
- ✧ 缆信息的显示
 - Cable Info: 显示缆信息
 - No Cbl Info: 不显示缆信息



下面来看一下它们对重复区域的影响, 如下图, 我们要创建和 RAZOR 装配相关联的重复区域, 对于 RAZOR 装配, 它由两个子装配组成, razor_body 和 razor_head, 而这两个子装配下还有它们自己的零件和子装配。看到这里, 不知道大家明白元件 component 和 part 之间的区别没? 其实很简单, component 是装配的一个组成部分, 它可以是零件 part 也可以是子装配, 对于 razor 来说, razor_body 和 razor_head 就是它的元件。而零件 part 就是指单一的文件。对于这个装配我们建立的重复区域和对应的报表符号如下图所示:



首先用 Flat, Duplicates 属性, 你就会得到如左下图的展开, 这是因为 Flat 属性只展开装配的下一级成员, 二 razor 的下一级成员只有 razor_body 和 razor_head, 所以展开后便只有这两项, 因为使用了 Duplicates 表明相同成员单独处理, 所以成员数目这栏没有意义为空。然后改为 Flat, Dup/Level 或 Flat, No Dup/Level 的话, 就会得到右下图的展开, 因为这时所有重复的元件当作一个处理并累加数目, 所以现在的元件数目就有意义了, 当然现在也只有一个。

2	RAZOR_HEAD	
1	RAZOR_BODY	
index	part name	qty

Duplicates

2	RAZOR_HEAD	1
1	RAZOR_BODY	1
index	part name	qty

No Dup/Level

一般来说，我们的重复区域更关心具体的零件 Part 的情况，这时就要把 Flat 属性改为 Recursive（递归）从而把所有等级的成员都展开以得到所有的成员。如左下图的 Duplicate, Recursive，你会看到，现在装配内的所有成员都显示出来了，包括所有的零件 part 和所有等级的子装配（如 razor_body, razor_head, razor 等，并且相同的零件都各自占了一行（如零件 5_296_026），自然这时元件数目也是没意义的。然后我们把 Duplicates 改为 No Duplicates，你就会看到表格大大缩短了，这是因为，所有相同的零件都变成了只占同一行，并且元件的数目累加。不过，你也要注意，这时的表格内同样还有所有等级的装配和子装配——毕竟这些也是 razor 的元件啊。

36	5_298_099	
35	5_296_047	
34	5_298_046	
33	5_296_026	
32	5_296_026	
31	5_296_026	

9	RAZOR_HEAD	
8	5_296_013	
7	5_296_012	
6	5_297_010	
5	5_298_003	
4	5_298_002	
3	5_298_001	
2	RAZOR_BODY	
1	RAZOR	
index	part name	qty

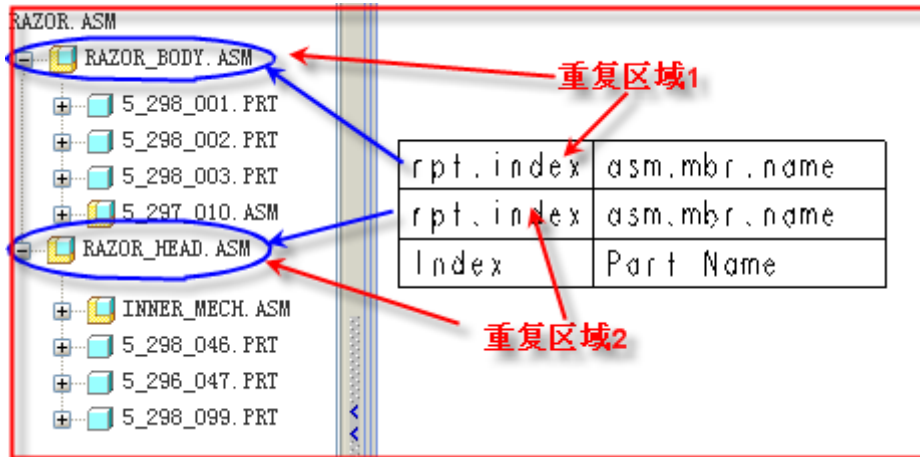
Duplicate, Recursive

13	RAZOR_HEAD	1
17	RAZOR_BODY	1
16	RAZOR	1
15	INNER_MECH	1
14	BLADES	1
13	5_299_027	9
12	5_298_099	1
11	5_298_046	1
10	5_298_033	2
9	5_298_013	1
8	5_298_010	1
7	5_298_001	1
6	5_297_010	1
5	5_296_335	3
4	5_296_047	1
3	5_296_026	16
2	5_296_013	1
1	5_296_012	1
index	part name	qty

No Duplicates, Recursive

我们在来看 Start Index 和 No Start Index 的应用。一个表中可以有多个并列的重复区域的。这样对于一个重复区域的索引就有两种情况了：要么使用自己的索引号从 1 开始不考虑别的重复区域（No Start Index），要么紧接着另一个重复区域继续索引下去（Start Index）。我们来看一个具体的例子：

如下图，我们的 Razor.asm 中有两个子装配 Razor_body 和 Razor_head。假设我们有个表中有两个并列的重复区域，分别关联这两个子装配（使用 Model/Rep 指定）。使用 Flat 属性。



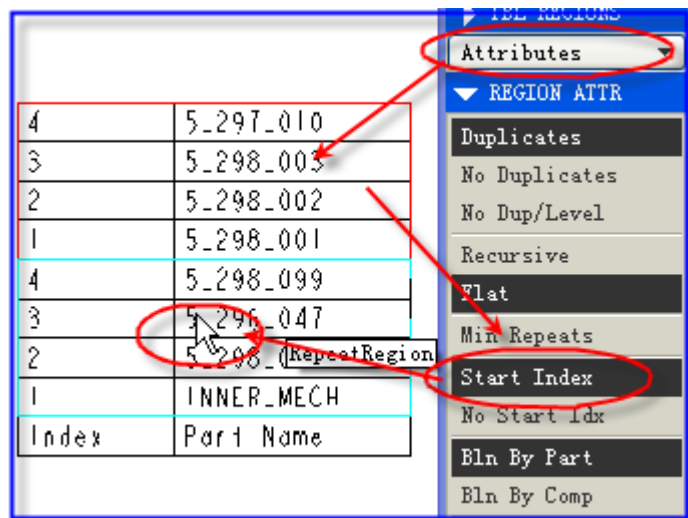
这样，你的重复区域现在的展开应该类似于左下图，注意到两个子装配的索引都是独立的从1开始（No Start Index），假设我们要改成右下图所示的展开，Razor_Body 的索引号紧接着 Razor_head 的索引号继续下去。

4	5_297_010
3	Razor_body
2	
1	5_298_001
4	5_298_099
3	
2	Razor_head
1	INNER_MECH
Index	Part Name

8	5_297_010
7	5_298_003
6	5_298_002
5	5_298_001
4	5_298_099
3	5_298_047
2	5_298_046
1	INNER_MECH
Index	Part Name

无维网Icefai原创ProE教程

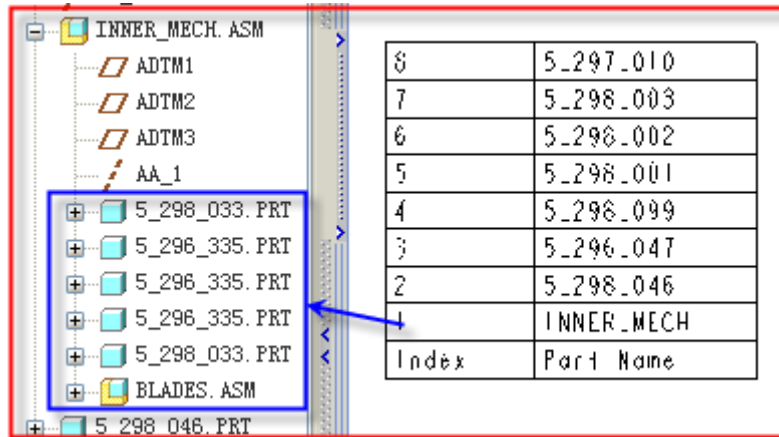
要达到这样的目的，你可以使用 Start Index 属性，首先进入 Attributes 选择 Razor_body 的重复区域，然后选择 Start Index 并选择 Razor_head 重复区域。这样新的重复区域展开就如右上图所示了。



Flat/Rec Item(平铺/递归 项目)

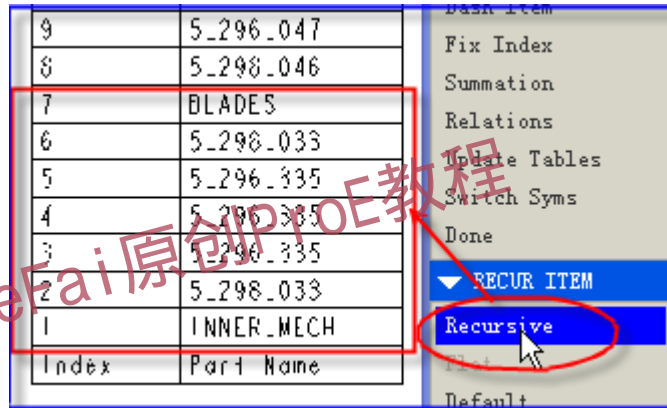
应用这个选项我们可以把重复区域内的某一项的属性改为平铺（Flat）或递归（Recursive）。这是因为在有的时候我们的某个子装配是外购的准件，所以我们需要把它当作一件在重复区域表中出现，但是重复区域中为了展开所有成员属性已经设为了 Recursive，这个时候就可

以用这个选项把外购件子装配的展开属性设为 Flat。而另一方面，有时候我们的重复区域的属性是 Flat 的但是对于某个子装配我们想单独展开，这时候我们就要把这个子装配的属性改为递归（Recursive）。如下图，我们沿用上一部的重复区域，重复区域的属性是 Flat，但是现在我们想展开 inner_mech 子装配。



这时我们就可以用 Flat/Rec Item 来把 inner_mech 的属性改为 Recursive，如右图所示。

现在 inner_mech 就展开了，不过只是展开下一级而不是展开所有成员，这点重复区域的 Recursive 的属性不同之处，使用时候注意。

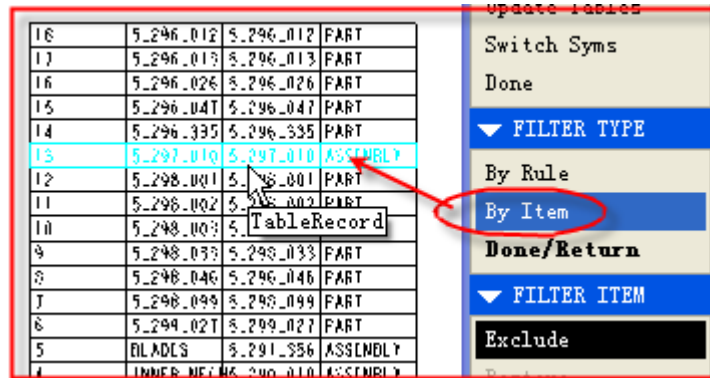


Filter(过滤)

细心的朋友或许注意到了，我们上面的例子中，重复区域中的成员都是有装配（Assembly）和零件（Part）混合显示的。一般来说，对于 BOM 表，子装配是不需要显示出来的。我们怎样才能让我们的重复区域中只显示类型为 Part 的成员呢？答案自然是使用 Filter(过滤项)来实现。使用 Filter（过滤）选项可以让你控制你的重复区域中的要显示的成员，这也是重复区域中的一个重要和常用的指令。

Filter 有两个选项：By Rule（按规则）和 By Item（按项目）。

使用 By Item 方法你可以直接手工选择要过滤掉的项目进行过滤。如下图，当然你也可以把过滤掉的项目重新显示出来。By Item 适合于过滤少量的特殊项目。



使用 By Rule 选项可以让你的过滤更智能化和一般化，过滤规则的表达式基本形式如下：

< Symbol> <Comparison operator> <literal value>

- <symbol> - 在重复区域有效的任何参数
- <comparison operator> - 任何一种运算符 <, >, <=, =>, == 和 !=
- <literal value> - 任何整数、浮点或字符串值

例如: &asm.mbr.type==part

需要注意的是规则在提示符行中的输入和编辑器中的显示不太一样的，在编辑器中的报表符号前没有“&”号，而在提示符行的输入必须带“&”。

过滤器从重复区域排除使用了任一个已过滤参数的记录以及与过滤器定义的约束不匹配的记录。系统省略表中的整个记录，而不只是由过滤器影响的参数。例如，重复区域的过滤器 &asm.mbr.type==part 将忽略对象的所有记录而不是报告中的零件。

注意：如果在两个方向过滤重复区域，则只删除特定单元，而不是整个记录。

过滤器，如 &asm.mbr.material!=steel，将从报告中排除所有带 "steel" 材料参数值（用户自定义）的组件成员的记录。

也可输入能对多个值进行操作的过滤器，例如 &asm.mbr.name==part_a,part_b,part_j，它将从报表中排除“part_a.prt”、“part_b.prt”和“part_j.prt”以外的所有组件成员记录。一行最多可包含 80 个字符。在创建带有多个可接受值的过滤器时，比如前面所述，只能使用运算符 "==" 和 "!="。

在重复区域使用带有这些运算符的过滤器时，应注意下列事项：

- 如果在重复区域向多个行添加带有 "==" 运算符的过滤器，所有条目将被遮蔽，且该表显示为空。
- 对带有多个值的运算符 "=="，这些值通过 "or" 连接。例如，对于 &asm.mbr.name==part_1,part_2,part_3，过滤器，重复区域要排除名为 "part_1"、"part_2" 或 "part_3" 零件以外的所有对象。
- 对带有多个值的运算符 "!="，该值通过 "and" 连接。例如，对于 &asm.mbr.name!=part_a, part_b, part_c 过滤器，重复区域将排除名为 "part_a"、"part_b" 和 "part_c"的零件。

比如对左下图的重复区域，我们现在要排除掉所有的 Assembly。那么我们只需要选择 By Rule 选项然后在提示符行输入下面的规则：

&asm.mbr.type==part

然后更新表格，你就会发现现在的重复区域中只剩下类型为 part 的成员了。

16	RAZOR_HEAD	5.294.010	ASSEMBLY
17	RAZOR_BODY	5.295.010	ASSEMBLY
16	RAZOR	5.296.001	ASSEMBLY
15	INNER_MECH	5.290.010	ASSEMBLY
14	BLADES	5.291.356	ASSEMBLY
13	5.299.027	5.299.027	PART
12	5.298.099	5.298.099	PART
11	5.298.046	5.298.046	PART
10	5.298.033	5.298.033	PART
9	5.298.003	5.298.003	PART
8	5.298.002	5.298.002	PART
7	5.298.001	5.298.001	PART
6	5.297.010	5.297.010	ASSEMBLY
5	5.296.335	5.296.335	PART
4	5.296.047	5.296.047	PART
3	5.296.026	5.296.026	PART
2	5.296.013	5.296.013	PART
1	5.296.012	5.296.012	PART
index	part name	part number	type

12	5.299.027	5.299.027	PART
11	5.298.099	5.298.099	PART
10	5.298.046	5.298.046	PART
9	5.298.033	5.298.033	PART
8	5.298.003	5.298.003	PART
7	5.298.002	5.298.002	PART
6	5.298.001	5.298.001	PART
5	5.296.335	5.296.335	PART
4	5.296.047	5.296.047	PART
3	5.296.026	5.296.026	PART
2	5.296.013	5.296.013	PART
1	5.296.012	5.296.012	PART
index	part name	part number	type

可以在报告过滤器中使用通配符 (*), 但是通配符只允许在使用运算符 "==" 或 "!=" 的过滤器中使用, 任何包含 "*" 的其它类型的过滤器都是错误的, 并且系统会提示重新键入过滤器。例如, 可将通配符用在以下过滤器中:

- &asm.mbr.name==part*
- &asm.mbr.name==*my*your*

第一个过滤器将匹配字符串 "part"、"part1"、"part_A" 和 "partabcdefg"。第二个过滤器会匹配字符串 "my"、"this_is_my_assembly"、"autonomy"、"not_yours" 和 "your"。

如果在过滤器的右侧添加一个反斜杠 (\), 系统照字面读取反斜杠后的字符 (象其自身一样), 而不作为特殊字符。然后, 可以过滤星号字符。例如, &asm.mbr.name>=part* 匹配大于或等于字符串 "part*" 的字符串。

Sort Regions (区域排序)

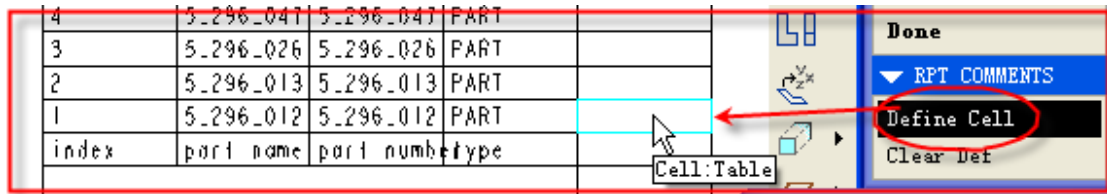
你可以根据选定的列举条目来 ASC 码的大小顺序来对重复区域进行排序。排序有两个选项: Forward 和 Backward, 分别表示顺序和倒序。排序比较简单, 要注意的是选择的时候是选择重复区域中的报表符号, 所以在有必要的时候使用 Switch syms 来显示报表符号。

Comments (注释)

注释单元是重复区域中的一个比较特殊的单元, 用户可以直接在这个单元中输入注释文本, 而不是从模型中读取数据。使用注释单元, 可注释重复区域中的一行数据, 并且用户的附加文本和该行保持关联, 即使该行的位置在区域内发生变化。

Pro/ENGINEER 跟踪注释单元至特殊模型 (而非参数值), 因此当模型名称改变时, 注释不会丢失。可以在除族表和缆线束以外的所有报告中使用注释单元。

定义 comment 的 cell 是注意选择的是重复区域报表符号所在的行上的单元格。



虽然注释可以和它同行的其它项目保持关联关系，但是一旦这行排除掉了（比如过滤，或重复区域属性更改导致）那么所加的注释便会丢失，即便以后这行重新恢复注释也不会随之恢复。

Indentation（缩进）

要理解缩进的用途，首先要理解一个概念，就是成员等级（Level），一个成员所在的装配中的等级是由它在装配中的“深度”决定的，也就是说从顶级装配经过多少重子装配才能到达它。比如定级子装配等级为 1，那么它的子装配等级就是 2，而子装配中的成员等级就为 3，以此类推。

缩进就是指下一级的成员在显示的时候要比上一级的成员缩进的字符数（类似模型树中的显示）。典型的缩进显示方式如下，你会看到装配内的成员都比装配本身右移一定数目的字符。

18	5.298.099	5.298.099	PART
17	5.296.047	5.296.047	PART
16	5.298.046	5.298.046	PART
15	5.296.026	5.296.026	PART
14	5.299.027	5.299.027	PART
13	BLADES	5.291.336	ASSEMBLY
12	5.296.395	5.296.395	PART
11	5.298.035	5.298.035	PART
10	INNER_MECH	5.298.010	ASSEMBLY
9	RAZOR HEAD	5.294.010	ASSEMBLY
8	5.296.013	5.296.013	PART
7	5.296.012	5.296.012	PART
6	5.297.010	5.297.010	ASSEMBLY
5	5.298.003	5.298.003	PART
4	5.298.002	5.298.002	PART
3	5.298.001	5.298.001	PART
2	RAZOR BODY	5.295.010	ASSEMBLY
1	RAZOR	5.298.001	ASSEMBLY
index	part name	part number	type

使用缩进的显示方式可以很直观的反映模型的装配结构，在有的场合还是很有用的。

要使用缩进(Indentation)，首先你的重复区域的属性必须是 Duplicates 或 No Duplicates/Level，不能是 No Duplicates，至于为什么不妨自己想想。想想如果同一个零件在两个以上的子装配中出现又会如何。

缩进的定义也简单，选择指令后再在重复区域中选择需要进行缩进排放的报表符号所在的单元格然后输入要缩进的字符数目就行了。

Dash Item（破折号项目）

可将绘图或报告中选取的 "rpt.qty" 和 "rpt.index" 值转换为破折号 "-"。在重复区域中，有的项目的索引或数目或许是没用的，这个时候我们就可以把它们转成“-“号。对于索引，原来它后面的索引号就会自动提前一位，如下图所示

9	RAZOR	Sort Regions
8		Comments
7		Indentation
6	5.2	Dash Item
5	5.2	Fix Index
4	ID3770 (NOTE)	Summation

7		5.296_0
6		5.296_0
-		5.297_010
5		5.298_003
4		5.298_002
3		5.298_001

如果要取消破折号，重新在使用一次就行了。

Fix Index (固定索引)

顾名思义，固定索引的意思就是把所选的项目的索引固定某个值。这也是手工排序的方法。实际工作中，利用重复区域自动生成的球标的排放可以说是没有顺序的。所以为了让我们的球标是按一定的方向来进行按索引号排序，也需要用到 Fix Index。

13	BLADES	Fix Index
12	5.296_0	FIX INDEX
11	5.298_0	Fix
10	INNER_WHEEL	Unfix
9	RAZOR_HEAD	Record
8	5.296_0	Index
7	5.296_0	Region
6	5.297_010	Done
5	Cell: Table 298.003	
4	5.298_002	
3	5.298_001	

选择指令然后选择重复区域内要固定索引的记录然后输入新的索引号就行了。更新后表格就会重排，并把记录固定到新的索引号上。要解除固定，可以用 Unfix 指令。要修改已经固定索引号的记录索引首先要将该记录的固定解除才行。

Summation (求和)

你可以利用重复区域的报表参数来进行累加。累加的选择方法有两个 By Name 和 By Text。使用 By Name 的话就会在点击后出现一个可以用来累加的报表参数，从这些参数中选择一个作为累加的参数就行。如果选择 By Text 就要直接在重复区域中选择报表符号所在的单元格。基本过程如下：

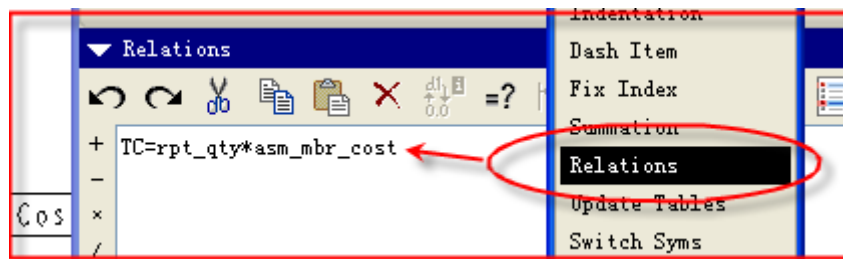
选择 Add, By Text, 然后选择一个报表符号所在的单元格。并在提示符行中输入新的求和的参数名。最选择一个用来放置求和结果参数的单元格便可

Total Cost	16.770
5.050	5.050
3.000	3.000
3.950	3.950
0.000	0.000
2.890	2.890
1.880	1.880
unit cost	total

Relations(关系)

你可以向重复区域中添加关系，并且关系中的参数也可以在重复区域中使用。关系的写法和一般的关系写法是一样的，并且还可以使用报表参数，注意的是报表参数和报表符号的区别，

报表符号中间是用点“.”。“连接的而报表参数是用”_“连接的。



上面的参数 TC，如果要在重复区域中使用可以用报表符号：&rpt.rel.(user defined)TC 来引用。这样就可以对重复区域中的数据进行一定的处理，比如求和，求差等。

Update Table 和 Switch Syms

重复区域的最后两个指令是 Update Table（更新表）和 Switch Syms（切换符号）分别是用来进行表的更新和报表符号的切换的，使用比较简单就不再说了。

嵌套重复区域

重复区域内可以创建新的重复区域，新的重复区域是原区域的子区域。在子区域中对父区域中的每个项目按照子区域的属性展开。

我们来看下面的表，表格中有三重重复区域，都是嵌套关系，重复区域的属性为 Flat

rpt.index	asm.mbr.name	asm.mbr.name	asm.mbr.name	asm.mbr.type
Index	Item Name		Type	

这个表更新后显示类似如下：

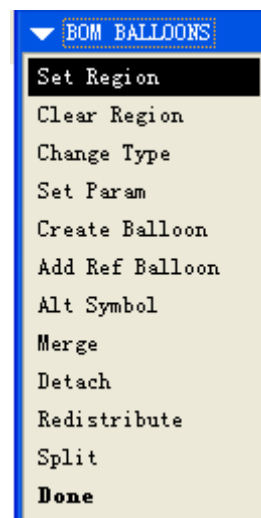
			BLADES	ASSEMBLY
			5_298_033	PART
		INNER_MLCH	5_296_335	PART
		5_296_099		
		5_296_046		
2	RAZOR_HEAD	5_296_047		
		5_296_003		
		5_296_002		
		5_296_001		
			5_296_013	PART
1	RAZOR_BODY	5_297_010	5_296_012	PART
Index	Item Name		Type	

我们来研究一下它的展开情况：首先这个因为这个重复区域的属性是 Flat，所以它的父区域只展开了下一级的成员，那就是 razor_head 和 razor_body 两个子装配。然后看一级子装配 razaor_head，你会看到在一级子区域中它也展开了它的下一级成员。二级子区域中同样也遵循了这个规律。这个的展开效果有点类似于采用缩进的方法来展开的效果。

BOM Balloons (BOM 球标)

重复区域定义好了之后，球标的创建就相对简单了。

- ✧ Set Region (设置区域) — 设定用来创建球标的区域
- ✧ Clear Region (清除区域) — 清除球标所关联的区域
- ✧ Change Type (改变类型) — 更改球标的类型
- ✧ Set Param (设置参数) — 设置用来显示在球标上的报表符号
- ✧ Create Balloon (创建球标)
- ✧ Add Ref Balloon — 添加参考球标
- ✧ Alt Symbol — 替换符号
- ✧ Merge (合并)
- ✧ Detach (脱离)
- ✧ Redistribute (重分配)
- ✧ Split (分割)



Set Region (设置区域)

要创建球标，第一步当然是为球标指定重复区域。在设置区域时你可以设定球标的类型，共有三个选项：Simple (简单)，With Qty (带数量) 和 Custom (自定义符号)。

Simple(简单):

圆内含有一个报表符号 (缺省的是 index 索引号)，
这个参数可以用 Set Param 来设定。

With Qty (带数量):

圆内在设定的参数下还有一个表示数量的报表参数

Custom (自定义):

用户自己定义的符号



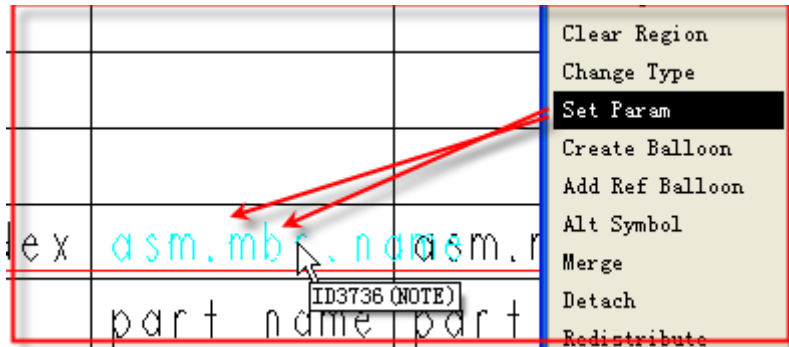
要想移除球标，你可以用 Clear Region 指令。

Change Type (改变类型)

当你的球标创建好之后，你可以随时更改它的类型，球标的类型如上所示。

Set Param (设置参数)

尽管一般来说，球标的报表符号都是索引号，但是这并不是必须的，你可以换成任何你喜欢的报表符号。比如 asm.mbr.name。如下图所示，选择 Set Param 指令后，首先选择重复区域，然后再选择要作为参数的报表符号，注意要在报表符号所在的单元格上选取。点 Ok 后你就会看到现在的球标圆内显示的就是装配的成员名字了。



Create Balloons (创建球标)

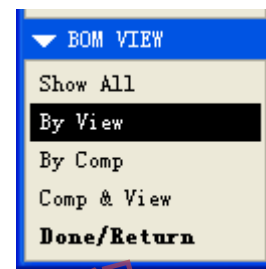
区域和参数(可选)设置好之后,你就可以创建球标了,选择指令后,系统会要求你选择球标的放置方法:

Show All (显示所有): 球标自动放置以显示出所有的球标

By View (按视图): 球标放置到所选的视图中

By Comp (按元件): 只生成所选的元件的球标

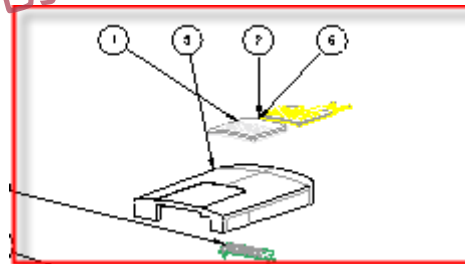
Comp & View (按元件和视图): 只生成在所选的视图所选的元件的球标



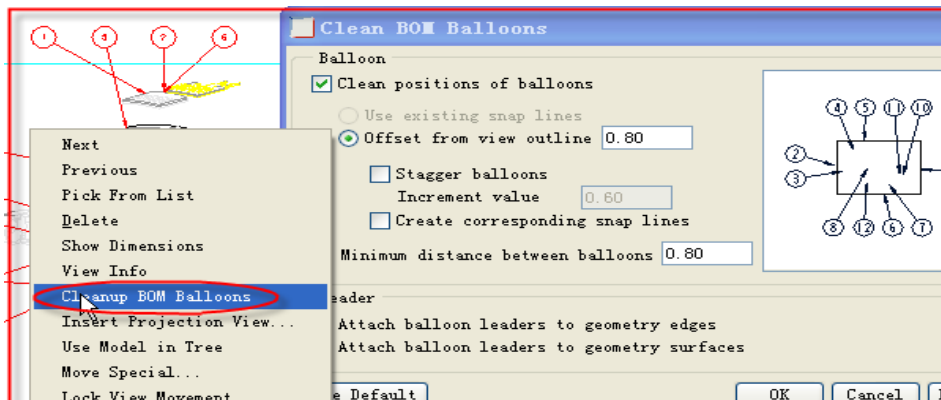
一般来说都是采用 By View 的方法在某个视图中放置的(通常是 Explode view)。点击创建指令后在要放置的视图上单击就可以实现球标的放置。

如右图所示为放置了球标后的视图。对于一个球标来说,它可以调整的地方有下面几个地方:

1. 球标大小
2. 球标的附着点形式
3. 球标的附着位置
4. 球标的位置
5. 球标的排列顺序



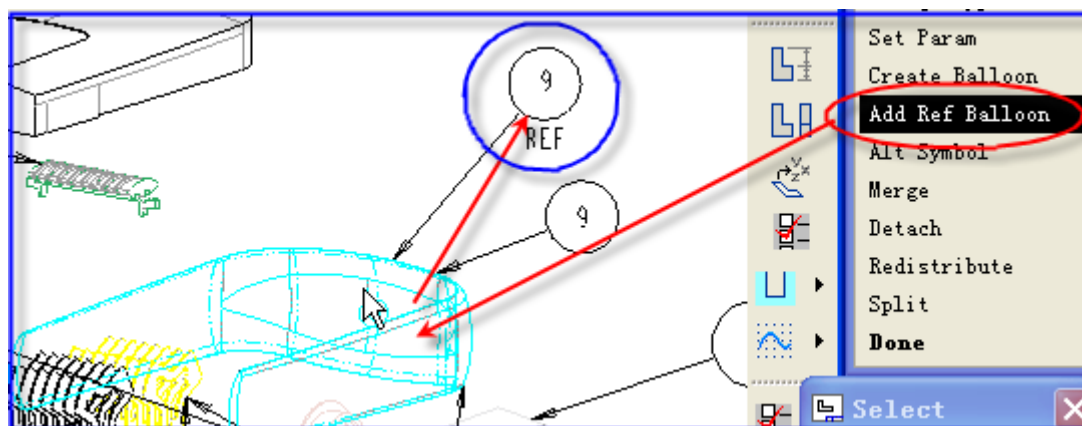
前面三项都是在 Drawing 的选项配置中设定的,详细的设定可以看附录中的选项说明。



球标放置好后,你可以在单击球标然后就可以按住左键随意移动球标的位置。也可以用 ProE 的 CleanUp (整理) 功能来进行自动的对齐,如上图所示。球标的排列顺序由重复区域中的 Fix Index 控制,可以看前面的详细说明。

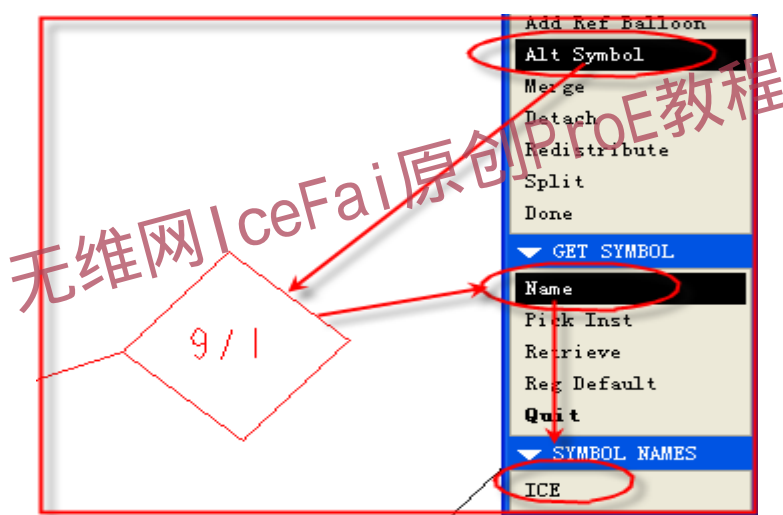
Add Ref Balloon (添加参考球标)

你可以为已经有球标的零件添加一个参考球标。如下图所示，选择指令后，选择一个已经有球标的零件最后中键放置参考球标的位置就行了。参考球标下面有个 REF 以表明它是参考的。



Alt Symbol (交换定制符号)

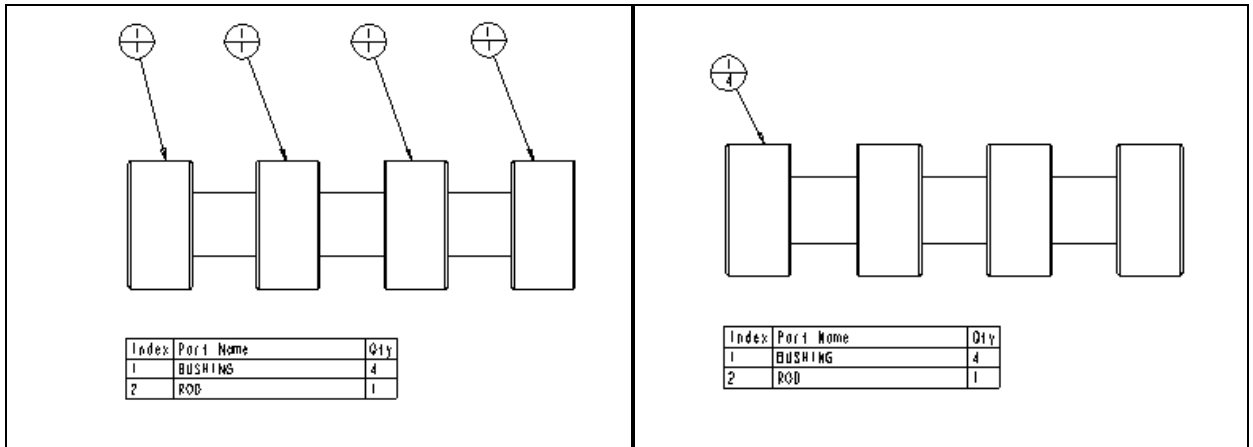
对于定制的球标符号可以随时进行更换



Merge(合并)和 Detach (分离)

可以将数量球标显示为一个显示元件数量部分总实例的球标， 或为每个元件实例显示一个球标。

如下图所示



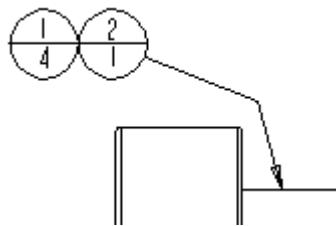
要合并数量球标（例如，要从左边的示例到右边的示例），

1. 单击“表”(Table) > “BOM 球标”(BOM Balloons)。菜单管理器打开。
2. 单击“合并”(Merge)。给出要求选取要合并的数量球标的提示。
3. 选取球标。给出要求选取要合并到的数量球标的提示。
4. 选取“主”球标。第一个球标被移动，主球标的数量加一。

合并命令仍激活，可以重复此过程。单击选取对话框中的“确定”(OK) 结束合并。

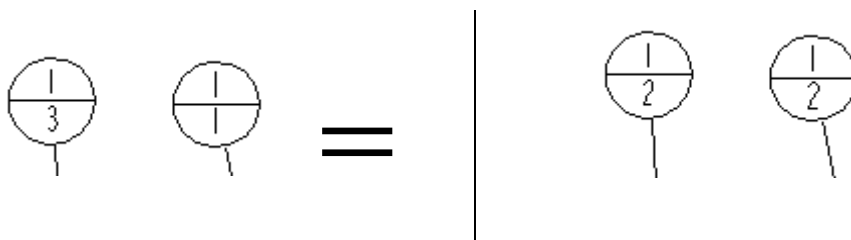
要反转该过程，请选择没有球标的一个元件，然后在鼠标右键弹出的快捷菜单中单击“创建 BOM 球标”(Create BOM Balloon)。如果需要，将提示要求给出所要数值的编号。（也可以在“BOM 球标”(BOM Balloons) 菜单管理器中使用“分割”(Split) 命令从合并球标创建新球标。）

堆放 BOM 球标



使用“合并”(Merge) 命令堆放具有不同索引号的数量球标。如果索引号不同，则球标水平邻接。使用“BOM 球标”(BOM Balloons) 菜单管理器中的“分离”(Detach) 命令分离堆放的球标。

重新分配 BOM 球标中的数量



1. 单击“表”(Table)>“BOM 球标”(BOM Balloons)...“BOM 球标”(BOM BALLOONS) 选项中的“菜单管理器”(Menu Manager) 打开。
2. 在“菜单管理器”(Menu Manager) 中单击“重新分配”(Redistribute)。
3. 选取数量大于 1 的 BOM 球标，如果对于重新分配数量有多种可能性，则给出执行步骤 4 的提示。否则提示选择“目标”球标。
4. 键入要从这个球标减去而添加到同一模型的另一个球标的数量。
5. 选取要添加数量的球标。球标数量就会相应地发生变化

无维网IceFai原创ProE教程

实例演示（一）

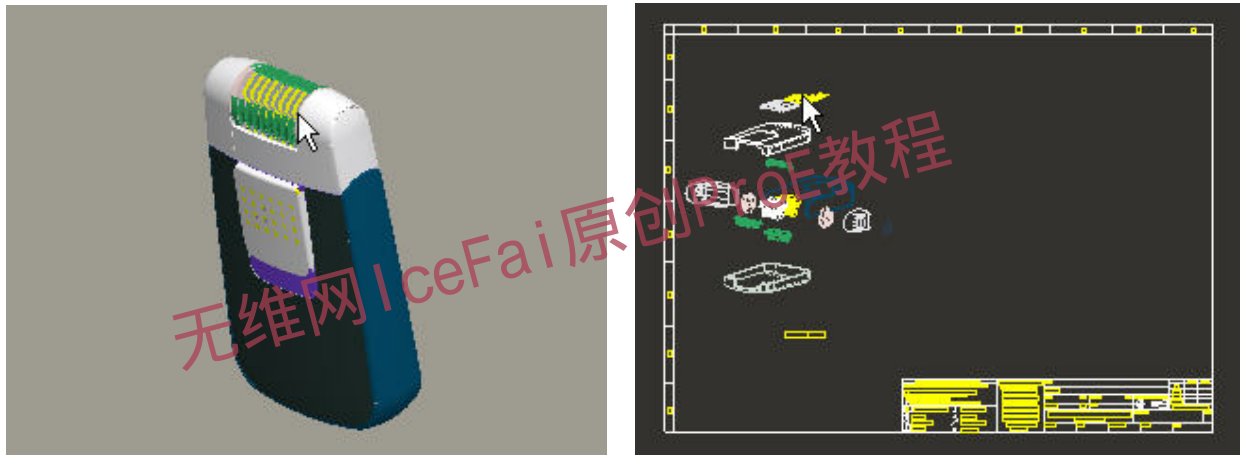
介绍

设置报表是个相对复杂的过程，但是一旦你定义好它们只后，你就可以保促起来并可以用到其它的 Drawing 上。

- 创建一个 BOM 表
- 添加表关系
- 表求和
- 添加过滤项
- 添加 BOM 球标
- 族表的报表

教程提供的样本文件

我们就用 ptc 提供的一个文件来进行讲解，例子用到的装配图如下所示：



分解装配 Drawing

在这步中我们将要在 Drawing 添加一个 Bill of Material (BOM) （材料明细表）以列出每个零件的有关信息。首先，我们需要创建一个有 8 列和 3 行的表格，按照下面的步骤来创建：

从 **Table** 菜单中选择 **Insert>Table**，然后选择 **Descending（降序）**，**Rightward（右向）** 和 **By Num Chars（根据字符数）** 作为创建选项。在屏幕上选择一个点，然后选择第一个 0 以使得每列有 10 个字符宽。重复这步直到生成了 8 列。所有 8 列完成后点鼠标中键接受。然后你就会看到右下图。对于每行的高度都选择 1，重复选 1 三次以创建一个 3 列，一个字符高的表。

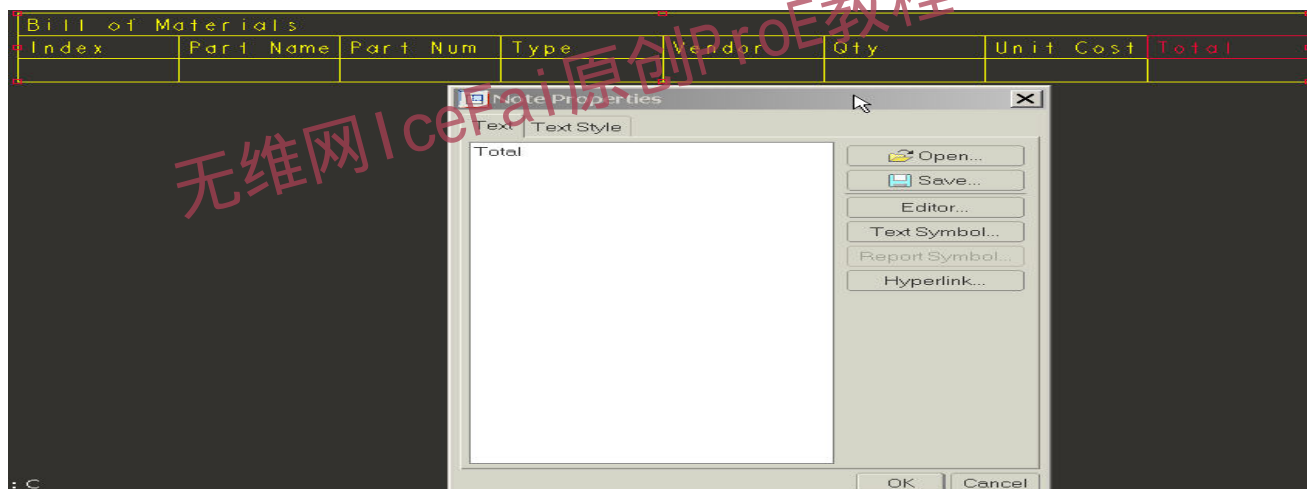


最后的结果如下图所示

合并表格单元

我们要合并最上面一列为一个单元格。首先从（表）**Table** 菜单中 选择 **Merge Cells** 然后选择第一行的第一格和第一行的最后一个格就行了。

然后在要作表头的表格中双击并输入合适的列表头。最后生成的表格如下图所示

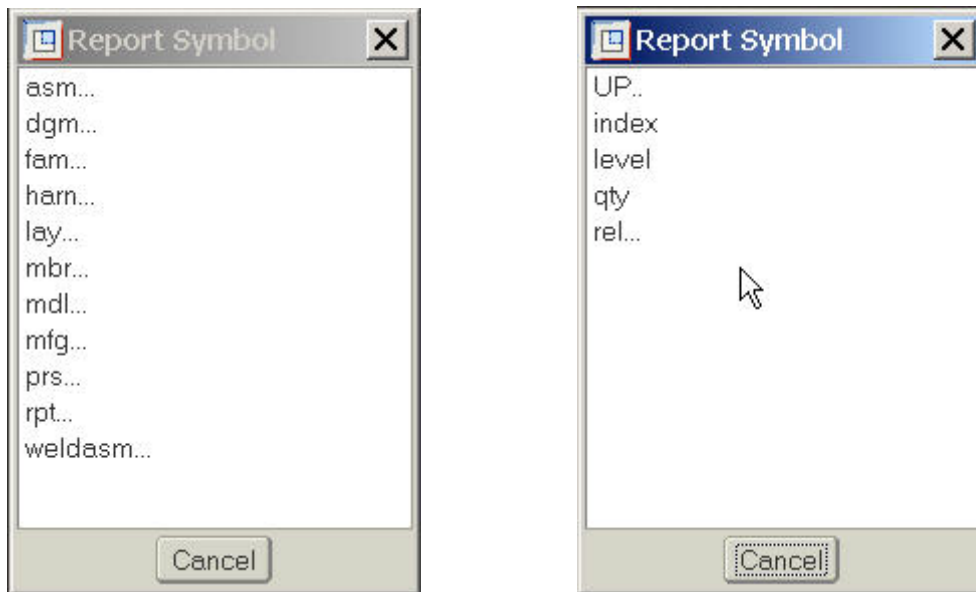


创建一个简单的重复区域（Repeat Region）

要创建一个简单的重复区域：从 **Table** 菜单中选择你 **Repeat Region**. 然后在菜单列表中选择 **Add**. 缺省地，重复区域的类型（**Region Type**）是为 **Simple**（简单）的。首先点击最后一行的第一格，然后再点击这行的最后一格就可以完成重复区域的角点定义，选择 **Done** 完成退出便完成重复区域的定义。

这样这个表格中最后一列中的表格单元中就可以用报表符号输入了。双击最后一行的第一个表格，系统就会弹出一个如左下图的报表符号列表框。你可以在这个列表中选择合适的报表符号。

然后选择 **rpt** 就会弹出如右下图的列表



接着在列表中选择 **index**.

重复上面的步骤，第三行中的表格单元分别用下面的报表符号来添填充。当要使用用户定义参数的时候选择 **User Defined**，然后在提示框中输入参数名就是了。

- asm.mbr.name
- asm.mbr.(User Defined) "Part_num"
- asm.mbr.type
- asm.mbr.(User Defined) "VENDOR"
- rpt.qty
- asm.mbr.(User Defined) "COST"
- 最后一个单元格暂时保留为空

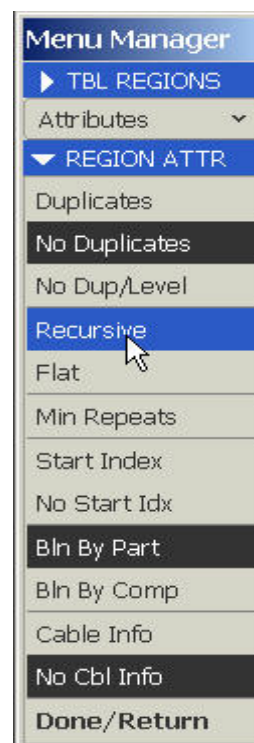
完成后，再生 **Regenerate**  > **Automatic** (自动)。结果应该和下表一致。

Bill of Materials							
Index	Part Name	Part Num	Type	Vendor	QTY	Unit Cost	Total
1	RAZOR_BODY	5_295_010	ASSEMBLY			0.000	
2	RAZOR_HEAD	5_294_010	ASSEMBLY			0.000	

重复区域属性

现在我们的表并没有显示出我们的装配中的所有零件。它仅是显示了装配中的顶级装配。造成这样的原因是因为在重复区域中的属性设为了 **Flat (平铺)** 而非**递归 (Recursive)**。并且这个表设为了重复(**Duplicates**)，这样在装配中的每一个相同的零件都被视为独立的零件，这样也造成了数量 (Quantity) 列的值为空。

要修改这些属性，从 Table 菜单中选择 **Repeat Region (重复区域)**。然后在弹出的菜单中选择 **Attributes (属性)** 在表中的任何地方点击一下以选中重复区域。将这个重复区域的属性设为 **No Duplicates (无重复)** 和 **Recursive (递归)** 然后点 **Done/Return** 退出。




报表关系

报表的最后一列中将添加一个关系用 quantity 乘以 unit cost 以得到准确的总费用。要实现这个功能我们需要添加一条关系。从 Table 菜单中选择 **Repeat Region (重复区域) > Relations (关系)**。然后在表中的任何地方选一个点以选中表，然后在弹出的菜单列中选择 **Add** 并输入下面的关系：

$$TC=rpt_qty*asm_mbr_COST$$

TC 是一个可变参数用来表示所有重复的零件的总费用。双击重复区域中能够的最后一列的第一个单元格并输入下列报表符号：

$$\&rpt.rel.(User Defined) "TC"$$

这样这个参数也包括了在重复区域中的参数，所以可以在重复区域上的每一行都计算都应的值。再生 Regenerate  **>Automatic (自动)**


Summation (求和)

在表格底部添加一行。从 Table 菜单中选择 **Insert>Row** 然后在最后一行底部附近点击一下。双击这行的倒数第二格然后输入 **TOTAL** 。

下面我们要在最后这行的最后一格创建关系以求出总费用。从 Table 菜单中选择 **Repeat Region (重复区域) >Summation (求和)**。在表中的任何地方点击以选中重复区域然后在弹出的菜单中选择 **Add (添加) >By Name (按名字)**。菜单下就会弹出一个可以求和的参数列表，选择 **TC** 来求和。



输入 **Total** 作为参数并点击最后一行最后一格作为放置表格。选择

Done/Return>Done 退出菜单。然后再生 Regenerate  **>Automatic (自动)** 以更新值。现在最后一格的值应该如下所示。

0.000	0.000
TOTAL	42.860

Filtering (过滤)

在这个表中同时有装配和零件，但可以通过使用过滤规则来限定只显示零件的。从 Table 菜单中选择 **Repeat Region (重复区域) > Filters (过滤)**。有两种方法可以选择 **By Rule (按规则)** 或 **By Item (按项目)**

By Item (按项) 可以直接移除你不想显示的项。选择 **By Item** 并直接点击第 6 项(part : 5_297_010) 然后中键接受。可以看到第 6 项被移除。

By Rule (按规则) 可以让你通过使用一条通用的并且应用于整个表的规则来移除不想要的项。在本例中，我们只想保留零件，所以可以选择 **By Rule (按规则) > Add (添加)**。然后输入下面的规则：

&asm.mbr.type== " part"

注意：规则表达式中有两个等号。

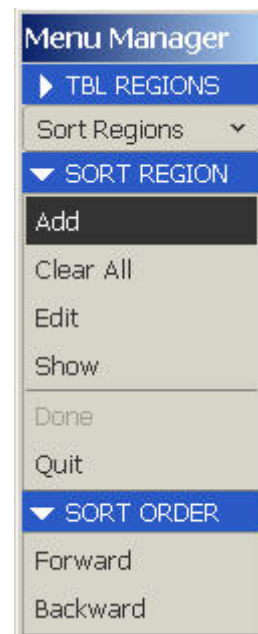
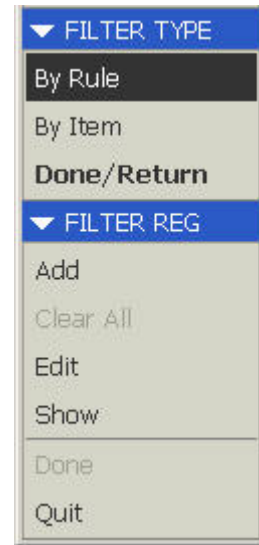
选择 **Done > Done/Return** 退出菜单你就会看到现在的表中只有零件了。

Sorting (排序)

表格可以安装阿拉伯字符来进行排序。

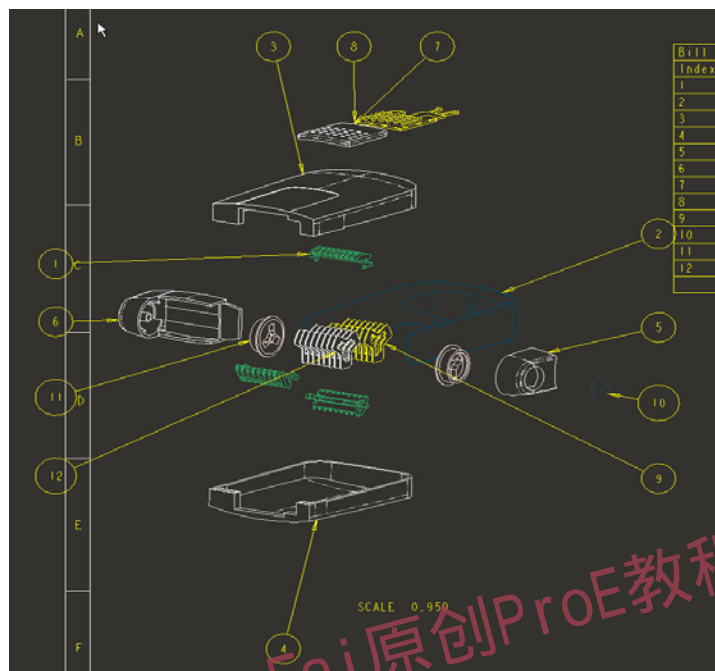
要进行排序，从 Table 菜单中选择 **Repeat Region (重复区域) > Sort Regions (区域排序)**。然后在表中的任意地方单击以选择重复区域，然后选择 **Add (添加) > Forward (向前)** 并在 Vendor 列中的第一个值上点击然后中键确认以退出选择。选择 **Done** 退出排序菜单，现在的表就是安字符排序了。

至此，表格我们就完成了。

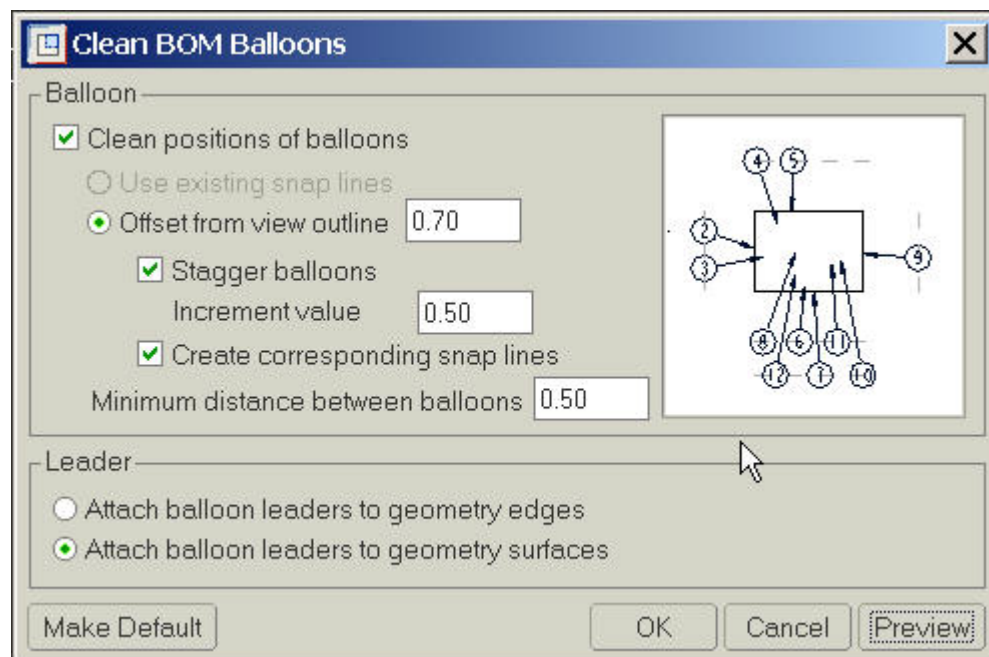


BOM Balloons (BOM 球标)

要自动创建球标,首先从 Table 菜单中选择 **BOM Balloons(BOM 球标)**. 当提示要 **Set Region (设定区域)**的时候在表中的任何地方单击一下就可以选择中区域。在本例中,我们只创建由索引号组成的 **Simple Balloons (简单球标)**。选择 **Create Balloon (创建球标)** 然后在分解视图上点击一下就可以, 现在的效果如下。



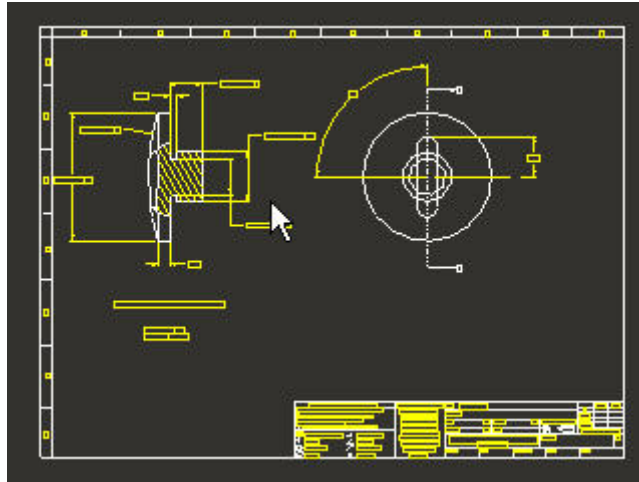
要整理球标的位置,首先选中视图然后选择 **Edit>Cleanup (清理)>BOM Balloons (BOM 球标)**系统就会弹出如下图的窗口, 点击 OK。



实例演示 (二)

TWO-D 重复区域的创建

打开文件 **fam_table.drw**.



首先创建一个 2X2 的表格。

然后在 Table 菜单中选择 **Repeat Region (重复区域) > Add (添加) > Two-D** 以创建一个 2D 重复区域。当提示 "Locate corners of the Region (选择区域的角点)" 时首先选择 A 和 D。然后在提示 "Select a cell to set the upper border of the row & column subregions" 时选择单元格 D。这样就创建了一个向下和向右展开的 2D 表格。对于不同的表格展开规则，按照下面的选择方法来进行选择：

A	B
C	D

Report Symbols (报表符号)

在表格中按照下图所示添加报表符号

- fam.inst.name
- fam.inst.param.value
- fam.inst.param.name

	fam.inst.param.name
fam.inst.name	fam.inst.param.value

再生 Regenerate  > **Automatic** (自动) 以更新值。 .

结果如下图所示：

	<i>grip_len</i>	<i>head_dia</i>	<i>head_rad</i>	<i>neck_dia</i>	SLOT	<i>thread_dia</i>
047_NS	3.000	12.000	20.000	3.400	N	4.600
047_32	3.200	12.000	20.000	3.400	Y	4.000
047_OH	5.000	15.000	25.000	3.400	Y	4.600
047_NN	8.000	12.000	20.000	4.000	N	4.000

无维网IceFai原创ProE教程

附录：有关重复区域和球标的 Drawing 配置选项

2d_region_columns_fit_text

确定二维重复区域中的每列是否自动调整大小以适应每列中的最长文本段，同时不会覆盖相邻的列或在表中形成较大间隙。

无文本的栏使用缺省栏宽（模板单元的宽度）。

缺省和可用设置

- **yes** - 调整 2-D 重复区域中每列的大小以使列中的最长文本符合列宽，并且不会和相邻栏重叠或在表中形成很大的间隙。
- **no*** - 列保持相同尺寸。

注意：对于包含自动调整的二维重复区域的表，不能手工调整其中的列。

● dash_supp_dims_in_region

控制尺寸值在 Pro/REPORT 表重复区域中的显示。

缺省和可用设置

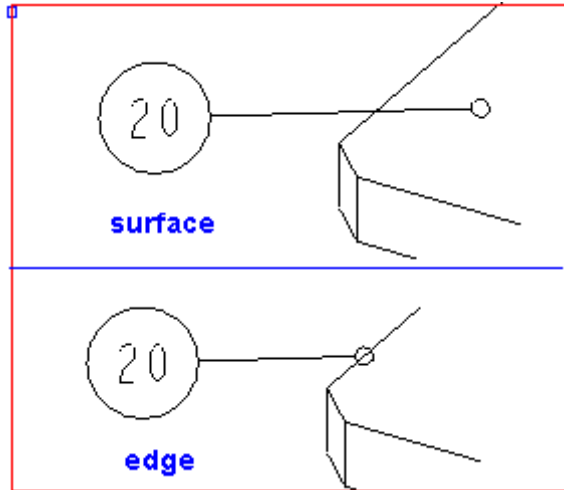
- **no*** - 在 Pro/REPORT 表重复区域中显示值。
- **yes** - 隐含尺寸而相应地显示短划线。

● def_bom_balloons_attachment

设定球标的引线附着零件的方式。两个选项 edge 和 surface。

缺省和可用选项值

- Edge*--球标的引线附在零件的边界上
- Surface—球标的引线附着在零件的表面。

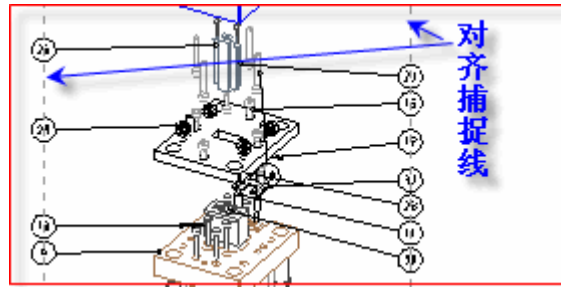


● def_bom_balloons_snap_lines

设定在创建球标的时候是否在视图周围创建辅助的捕捉线。

缺省和可用设置

- No*---不创建
- Yes---创建



● def_bom_balloons_stagger

设定球标位置是否在捕捉线附近摆动

缺省及可用值

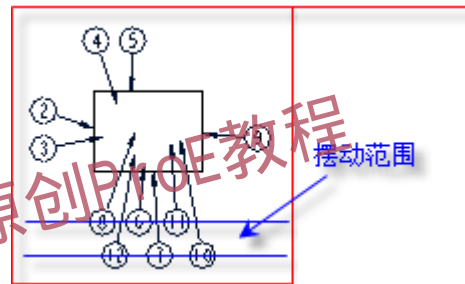
- no*--
- yes

● def_bom_balloons_stagger_value

设定球标位置的摆动幅度

缺省及可用值

- 0.6*
- 值

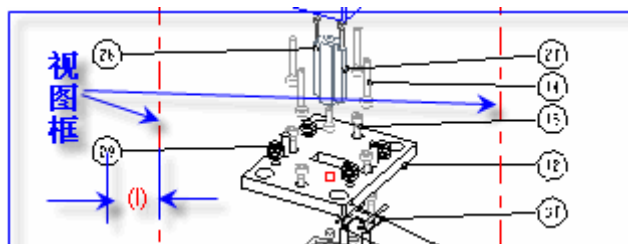


● def_bom_balloons_view_offset

设定球标偏离视图框的边界的距离大小，缺省为 0.8 (inch)

缺省及可用值

- 0.8*--
- 值--



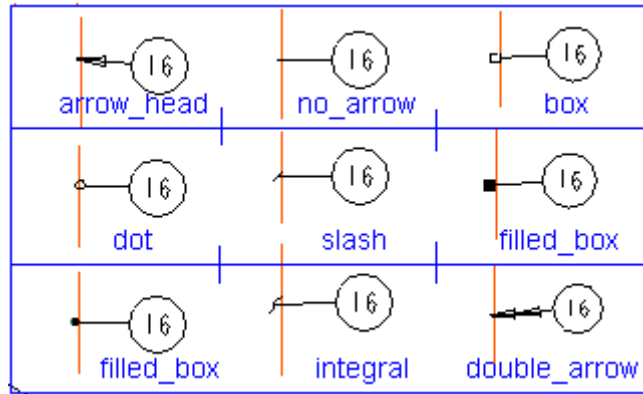
● def_bom_balloons_edge_att_sym

设定球标引线在零件边界上的附着点的显示方式

缺省及可用值

- arrowhead*--单箭头
- dot-----小圆圈
- filled_dot--实心小圆圈

- no_arrow—无附着点形式
- integral—积分号，波浪线
- slash---斜线
- box—正方形
- filled_box—实心正方形

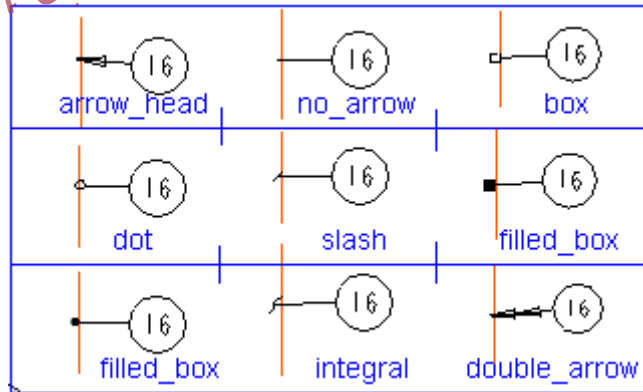


● def_bom_balloons_surf_att_sym

设定球标引线在零件表面上的附着点的显示方式

缺省及可用值

- arrowhead--单箭头
- dot-----小圆圈
- filled_dot--实心小圆圈
- no_arrow—无附着点形式
- integral*—积分号，波浪线
- slash---斜线
- box—正方形
- filled_box—实心正方形

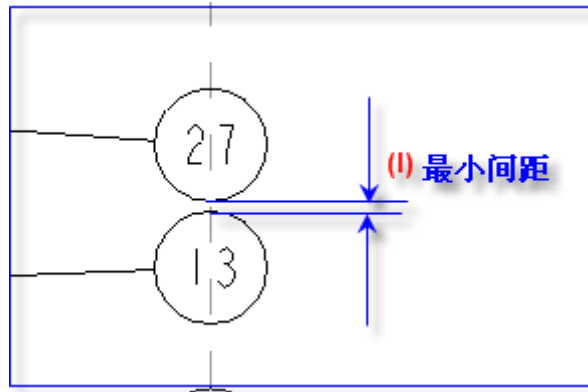


● min_dist_between_bom_balloons

设定两个球标之间的最小距离.

缺省及可用值

- 0.8*--
- 值



● model_digits_in_region

控制二维重复区域中小数位数的显示。

缺省和可用设置

- **yes*** - 二维重复区域将反映零件和组件模型尺寸的数字位数。
- **no** - 二维重复区域数字独立于零件和组件模型尺寸设置。

● reference_bom_balloon_text "DEFAULT"

设定参考球标的缺省指示文本

缺省及可用值

- “DEFAULT”--缺省值，参考的球标后面用“ref”来代替数量值（如果有的话）
- 文本--

● show_cbl_term_in_region

对于有终结器参数连接器的电缆组件，可用报告符号 `&asm.mbr.name` 和 `&asm.mbr.type` 在 Pro/REPORT 表中显示终结器。

缺省和可用设置

- **no*** - 对于现有绘图，缺省值为 no。
- **yes** - 显示终结器。必须为重复区域设置属性“缆信息”(Cable Info)。创建新绘图时，缺省值为 yes。

● sort_method_in_region

设定在重复区域内项目的排序方法

缺省及可用值

- delimited*--用分隔符来确定
- string_only--用名字的字母顺序来确定

- trailing_numbers—用尾随的数字来确定
- proe_2001—用 2001 的排序方法来进行

- **zero_quantity_cell_format**

设定在重复区域中对于零数目的零件的表示方法

缺省及可用值

- empty*--不显示任何指示
- -
- 0

无维网IceFai原创ProE教程