



极限切割 2006

用户手册

感谢您购买使用黑格软件产品！

在使用本产品前，请仔细阅读下面的许可协议。

同意本许可协议的所有条款及此处包含的任何补充或特殊的许可条款是获得本产品许可的必要条件。如果您不同意此协议的所有条款，请在三天内将产品退还济南黑格软件有限公司或其代理商。您对本软件的使用将表明您同意接受本协议中条款的约束。

许可协议

授予使用许可权

您可以为了备份的目的而复制光盘中的软件。

禁止使用

除按上述“授予使用许可权”条款被授权的行为外，不可以复制、修改、逆向工程、分解或重组该产品的全部或部分，不可向他人销售、租借、许可、转让、分发全部或部分本产品或本协议授予的权利。

有限担保

保证在自产品交给您之日起的 30 天内，在正常使用情况下，产品不会出现实质性的质量问题和生产设计上的缺陷。自您购买产品之日起的 30 天内，济南黑格软件有限公司的全部责任和您能获得的全部补救措施为：可选择退货、更换或修理；购买产品之日起 30 天以后，您能获得的全部补救措施为：更换或修理。

除了上述对本产品的原始购买者所提供的有限保证之外，不向任何人作任何其它的保证。对济南黑格软件有限公司的产品、性能或服务亦没有明示的或暗示的或其它任何形式的保证，包括但不限于商品的适销性和对特定用途的适用性。

任何情况下，无论如何引起及依据何种责任理论，均不承担任何因使用或不能使用本产品造成的损失责任，包括：由于使用本产品或本产品的功能失效导致丢失数据、损失利润及其它特别的、偶然的、附随的、继发的或间接的损失。

产品设计变更

本文档发行之后产品设计可能发生变更，不能保证文档内容和产品设计完全一致，济南黑格软件有限公司也不会因此而另行通告。

所有权

所有的产品，包括软件狗、软件、文档、与本产品一并附送的其它材料及您制作的备份的所有权与版权均属于济南黑格软件有限公司。

协议的终止

违反上述条款时，本协议的授权将自动终止。

Queek（黑格）是济南黑格软件有限公司注册商标。本文所涉及的其它产品和公司名称可能是各自相应所有者的商标。

目录

1	序言.....	1
1.1	打开产品包装.....	1
1.2	安装驱动程序.....	1
1.3	安装应用软件.....	2
1.4	如何获得技术支持.....	2
2	入门操作.....	3
2.1	打开演示任务.....	3
2.2	计算.....	4
2.3	查看汇总信息.....	5
2.4	打印.....	5
2.5	输入原材料.....	5
2.6	输入零件清单.....	6
2.7	手动排样.....	6
3	任务属性.....	7
3.1	默认材料类型.....	8
3.2	默认切割参数.....	8
3.3	度量单位定义.....	8
4	原材料与零件编辑操作.....	8
4.1	打开编辑器.....	8
4.2	列操作.....	9
4.2.1	设置表格属性.....	9
4.2.2	编辑器选项.....	10
4.2.3	原材料属性.....	11
4.2.4	零件属性.....	11
4.2.5	改变列的位置和宽度.....	12
4.2.6	线材设置.....	12
4.2.7	板材设置.....	12
4.2.8	卷材设置.....	13
4.2.9	坯材设置.....	13
4.3	行操作.....	14
4.3.1	添加行.....	14
4.3.2	插入行.....	14
4.3.3	移动行.....	14
4.4	复制与粘贴.....	15
4.4.1	把数据导出到 Excel.....	15
4.4.2	从 Excel 导入数据.....	16
5	自动计算.....	17
5.1	设置原材料策略.....	17
5.2	设置零件策略.....	17
5.3	设置余料策略.....	18

5.4	设置搜索策略.....	19
6	手工排样.....	19
6.1	添加作业.....	19
6.2	删除作业.....	20
6.3	把原材料拖放到作业区.....	20
6.4	把零件拖放到作业区.....	20
6.5	转动零件.....	21
6.6	零件分组.....	21
6.7	拆解零件组.....	23
7	打印.....	23
7.1	选择当前预览的内容.....	23
7.2	显示零件组细节.....	23
7.3	多页布局.....	24
7.4	打印机设置.....	25
7.5	文档设置.....	25
7.6	打印输出.....	26
8	高级技巧.....	27
8.1	撤销与重做.....	27
8.2	使用“无限制”作为原材料数量.....	27
9	安装插件.....	27
9.1	插件类型.....	27
9.2	添加插件.....	28
9.3	删除插件.....	28
9.4	标准插件.....	28
10	开发商技术指南.....	29
10.1	系统架构.....	29
10.2	如何开发插件.....	30
10.3	*.xct 文件格式.....	30
10.4	xcuttask.dll.....	32
10.5	xcutedit.ocx.....	38
10.6	xcutcalc.exe.....	40
10.7	xcutprnt.exe.....	40
10.8	import05.exe.....	40
10.9	export05.exe.....	40
10.10	import05.exe 和 export05 的主要算法源代码.....	40
11	售后服务和技术支持.....	49
11.1	产品更新.....	49
11.2	联系方式.....	49

1 序言

《极限切割 2006》可用于线材、板材、卷材、坯材的切割优化计算。

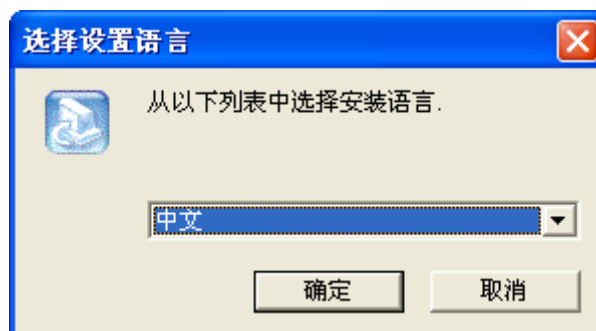
1.1 打开产品包装

本产品包括如下物品：

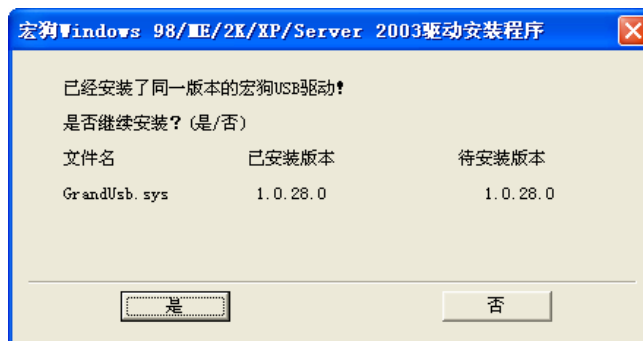
安装光盘一张
软件狗一枚
产品合格证一份

1.2 安装驱动程序

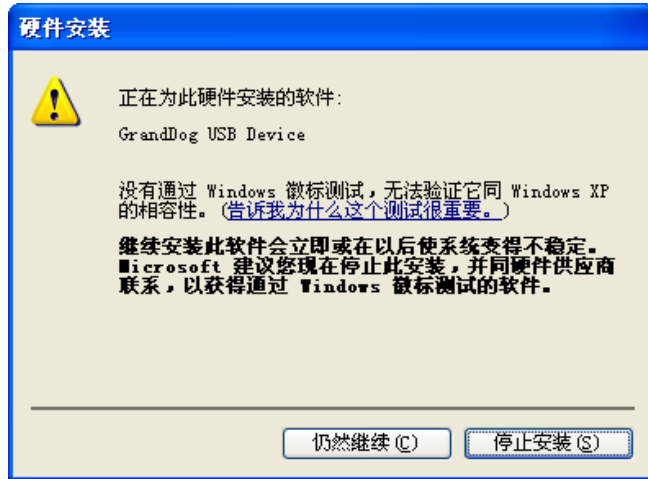
打开安装光盘，运行 DogSetup.exe 程序。选择“中文”



如果您曾经安装过驱动程序，系统会出现如下提示：



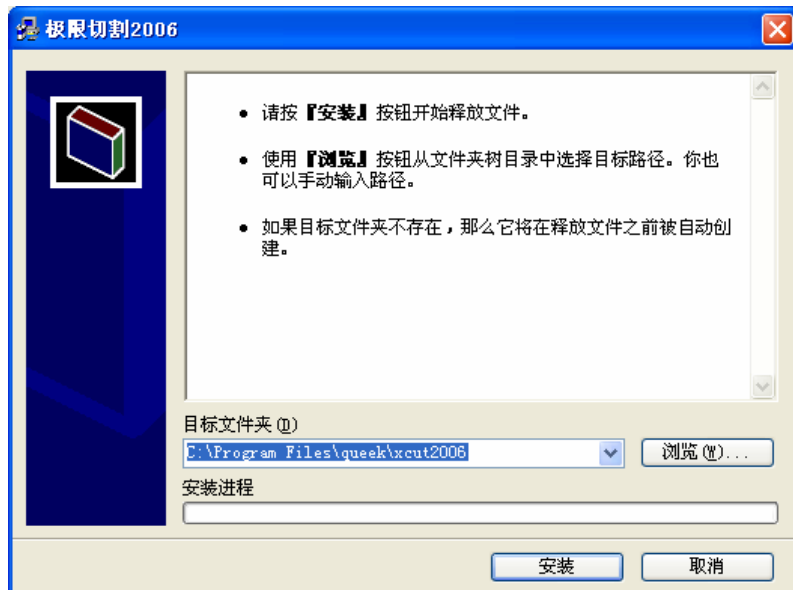
如果软件狗已经插入计算机的 USB 口，Windows 会出现提示：



选择“仍然继续”，完成驱动程序安装。

1.3 安装应用软件

打开安装光盘上的 Setup.exe 程序，选择安装位置：



选择“安装”，完成安装过程。您可以在 Windows 桌面或者“开始”菜单里找到“极限切割 2006”应用程序。

1.4 如何获得技术支持

您可以与黑格客户服务中心联系获取相关技术支持。联系方式如下：

黑格客户服务中心：

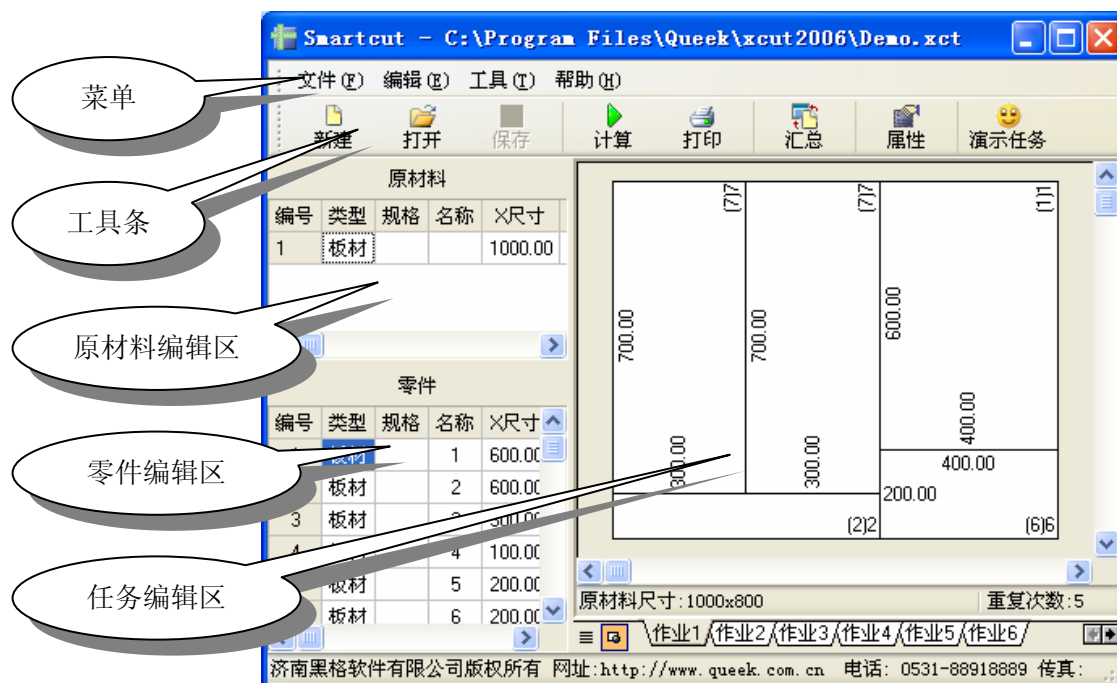
电话：0531-88918889

传真：0531-8891-8886

邮件：kefu@queek.com.cn

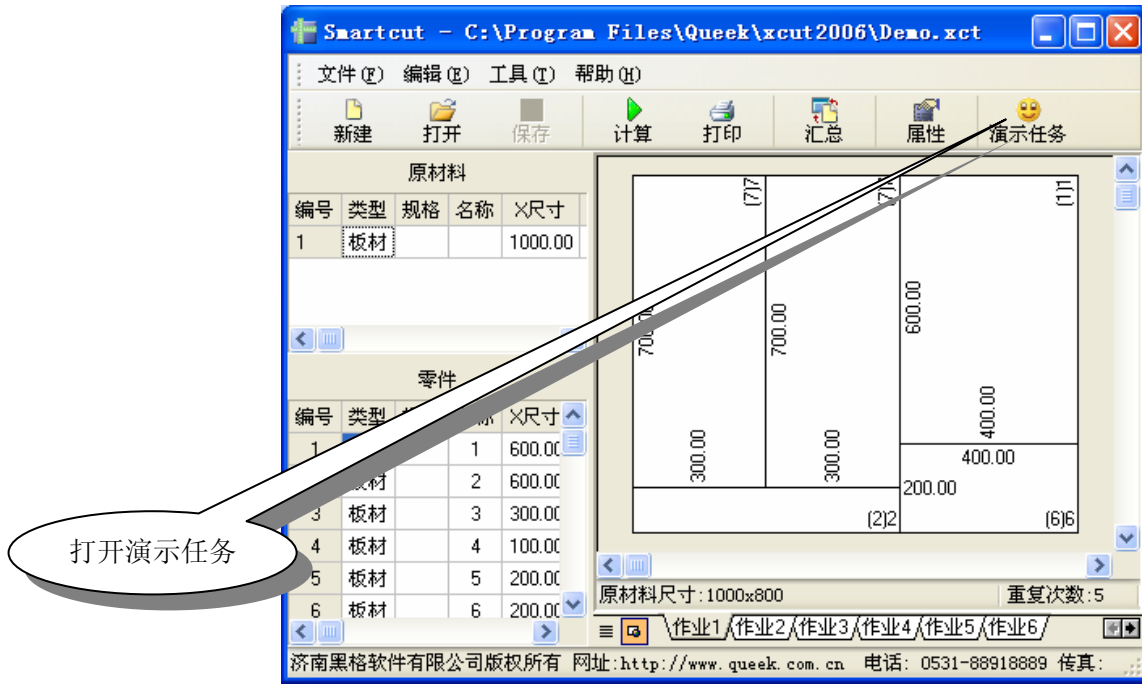
2 入门操作

《极限切割 2006》运行后，可以看到程序主窗口如下：



2.1 打开演示任务

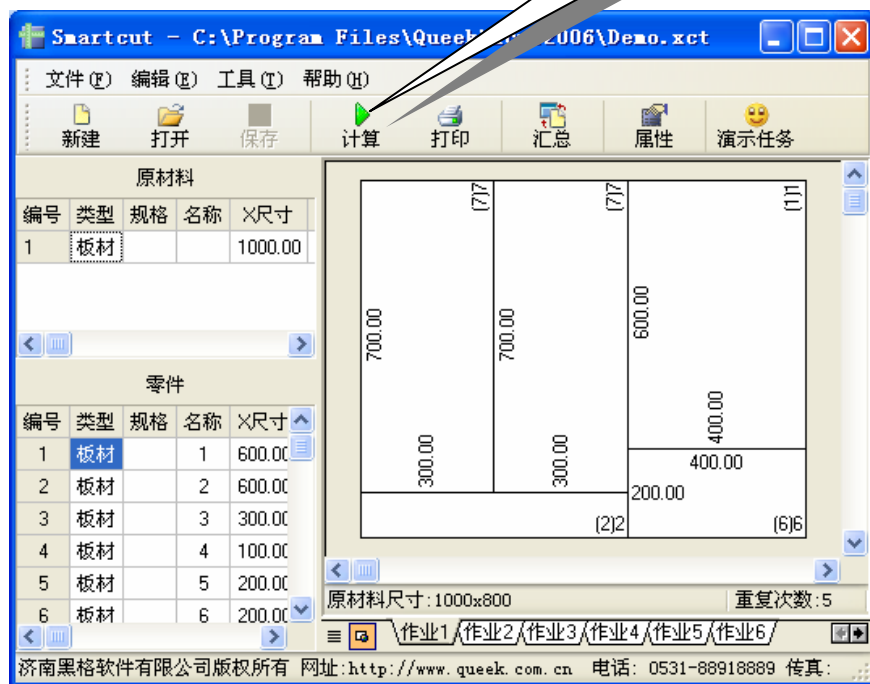
点击工具条上的“演示任务”。



演示任务提供了原材料和零件的基础数据，可以先利用这些数据练习《极限切割 2006》的基本操作。

2.2 计算

点击工具条上的“计算”按钮，如下图：



软件显示计算对话框：



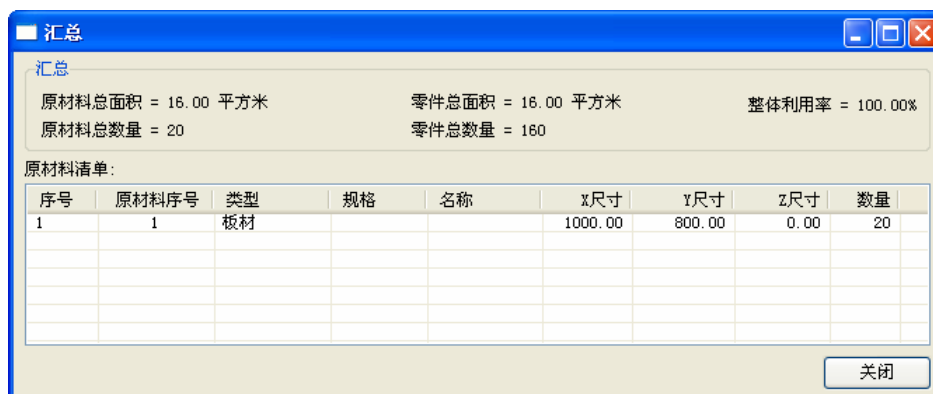
点击“开始”按钮，即可完成计算。

开始计算

注意：这个切割任务的整体优化率是 100%，这是《极限切割 2006》通过的 100 分性能测试的内容之一。

2.3 查看汇总信息

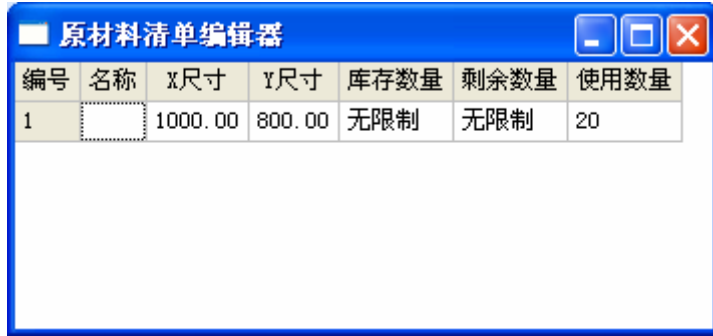
点击工具条上的“汇总”按钮，即可显示本次任务的汇总信息。



2.4 打印

2.5 输入原材料

双击原材料编辑区，可以打开原材料编辑窗口。



您可以在此编辑窗口录入数据，每个数据项填写完毕，可以直接键入回车键把光标移入下一个编辑单元格。

2.6 输入零件清单

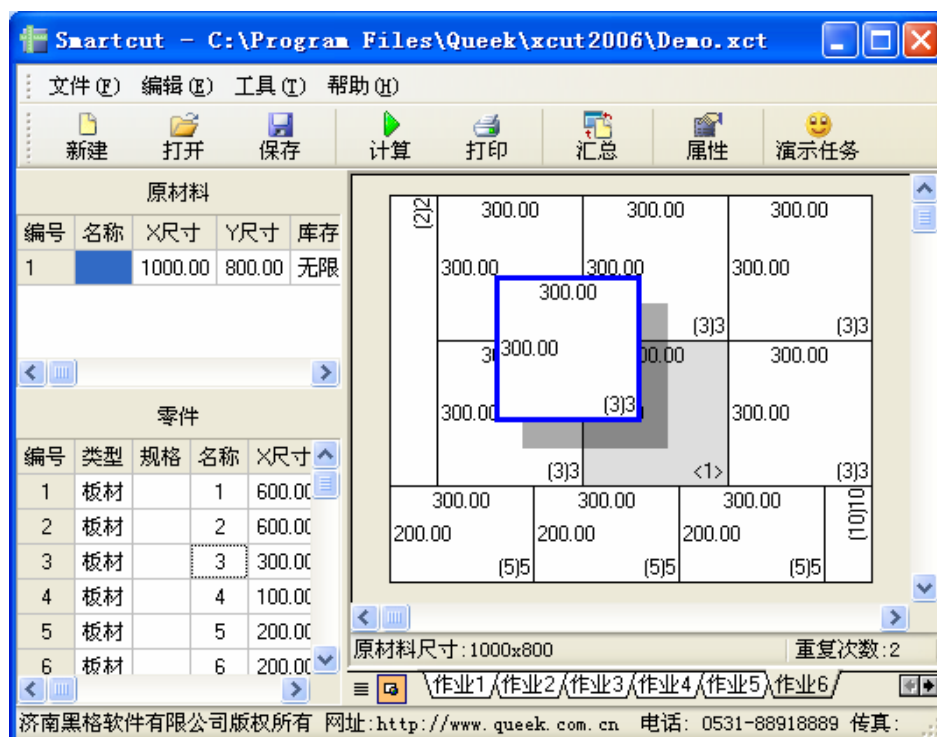
双击零件编辑区，可以打开零件编辑窗口。



您可以在此编辑窗口录入数据，每个数据项填写完毕，可以直接键入回车键把光标移入下一个编辑单元格。

2.7 手动排样

《极限切割 2006》提供了手工排样操作，你可以尝试用鼠标拖动任务编辑区的零件。参见下图：

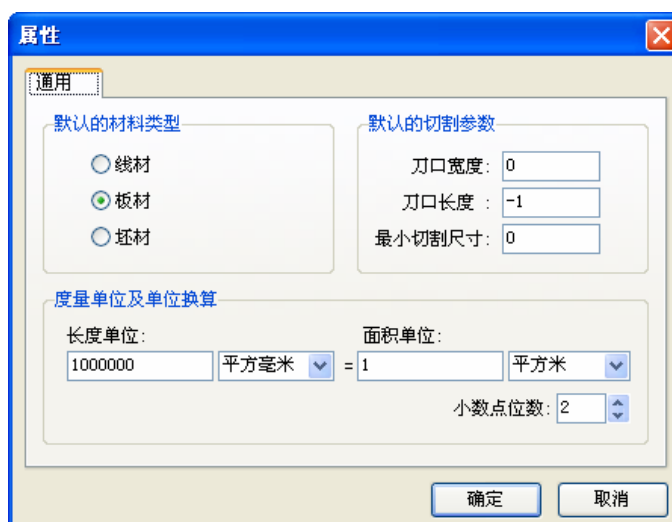


您可以把任务编辑区的零件拖放到零件编辑区，这样就把排样图中的零件删除了。

您也可以通过把零件编辑区的零件拖放到任务编辑区，这样可以把零件加入当前任务。

3 任务属性

《极限切割 2006》提供了许多用户定义的数据项，这有可能增加数据录入的复杂性。我们可以通过预定义任务的属性参数，简化数据的录入操作。属性录入窗口如下：



3.1 默认材料类型

《极限切割 2006》允许线材、板材、坯材三种类型的材料。设置默认的材料类型后，在创建新的原材料时，会首先选择默认材料。

3.2 默认切割参数

默认的切割参数有三个：刀口宽度、刀口长度和最小切割尺寸。

- 刀口宽度：指切割的损耗宽度。设置为 0 时表示没有损耗。
- 刀口长度：表示切割的最大长度。例如剪板机的最大切割长度是有限的。设置为-1 表示切割长度没有限制。
- 最小切割尺寸：由于工艺的原因，某情况下零件或余料小于某个尺寸时，切割就无法进行了。例如玻璃切割的掰边距、某些特殊材料切割的工装夹具预留位置等。该数值设置为 0 表示没有最小切割尺寸的限制。

3.3 度量单位定义

度量单位定义包括两部分内容：单位名称的选择和精度的定义。

- 度量单位可以自己定义。上面图中的例子表明，长度单位我们选用“毫米”，面积单位我们选择的是“平方米”，二者的换算单位是：1000000 平方毫米=1 平方米。
- 小数点位数可以自己选择。图中例子选择保留 2 位小数点。

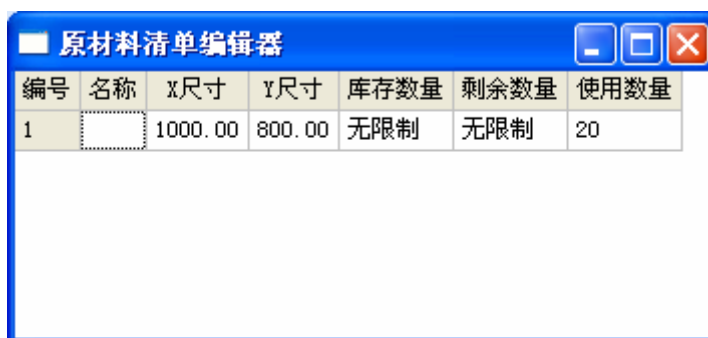
注意：《极限切割 2006》在内部采用的整数保存数据，这样就不存在数值计算中常见的舍入误差问题。

4 原材料与零件编辑操作

《极限切割 2006》允许在主窗口直接编辑原材料和零件数据，也可以通过双击编辑区打开专用的编辑窗口。专用的编辑窗口提供了更为方便的数据编辑功能。

4.1 打开编辑器

我们以原材料编辑窗口为例子，在原材料编辑区双击鼠标，打开原材料编辑窗口：



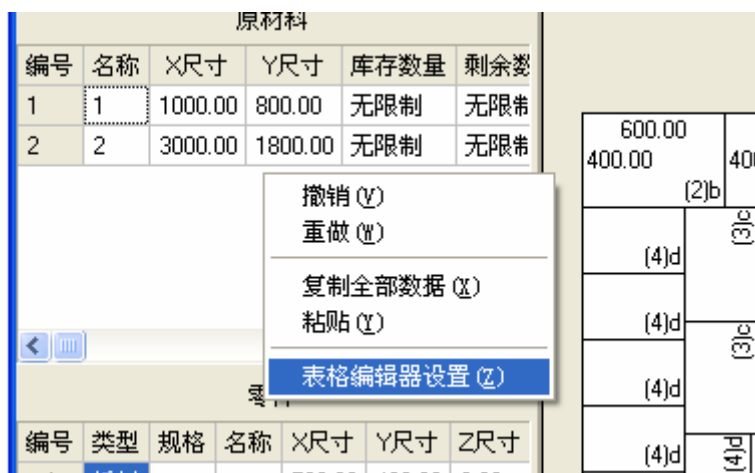
我们看到一个简单的编辑窗口。实际上，这个编辑窗口有许多方便操作的功能。下面我们详细给予介绍。

4.2 列操作

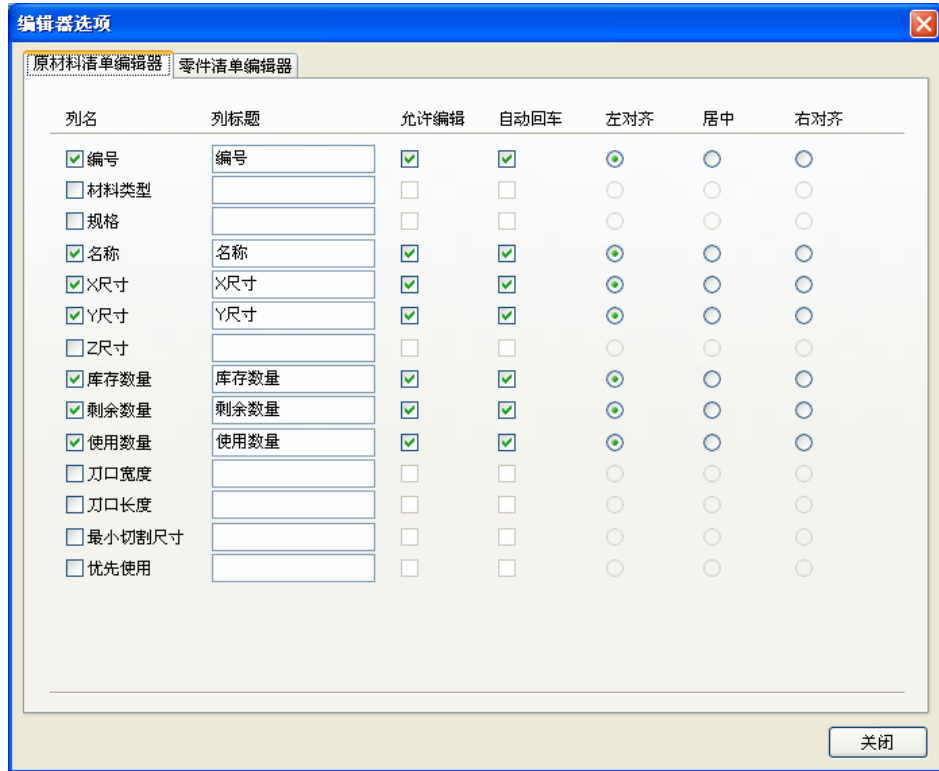
在上面的窗口中，编辑区只包含 7 列数据。实际上，《极限切割 2006》允许编辑更多的列。本小节我们介绍编辑窗口有关于列的操作。

4.2.1 设置表格属性

在编辑窗口单击鼠标右键，弹出右键菜单。参见下图：



选择最后一项“表格编辑器设置”。参见下图：



我们可以看到这是一个两页内容的设置窗口。第一页“原材料编辑器”提供了原材料编辑器的基本设置，第二页“零件清单编辑器”提供了零件清单编辑器的基本设置。下面我们详细介绍这两页内容。

4.2.2 编辑器选项

编辑器有如下选项：

名称	意义
列名	列的内部名称
列标题	在屏幕上显示名称
允许编辑	是否允许修改该列的内容
自动回车	当键入回车键时，是否允许光标自动移到本列
左对齐	采用左对齐方式显示单元格内容
居中	采用居中齐方式显示单元格内容
右对齐	采用右对齐方式显示单元格内容

4.2.3 原材料属性

原材料属性详见下表：

属性名称	意义
编号	原材料的序号
材料类型	线材、板材、坯材三种类型之一
规格	材料的规格型号。原材料被用于加工某种零件的前提条件是：材料类型和规格都和零件一致。
名称	原材料的名称。两个原材料同名的条件是，这两种原材料具有相同的材料类型、规格和 X、Y、Z 尺寸。在许多场合，建议采用物料编码做原材料的名称。
X 尺寸	X 尺寸，-1 代表无限制。
Y 尺寸	Y 尺寸，-1 代表无限制。如果原材料是卷材，可设置 Y 尺寸 = -1。
Z 尺寸	Z 尺寸，-1 代表无限制。
库存数量	当前库存数量，-1 代表无限制。
剩余数量	完成本次任务后的库存数量，-1 代表无限制。
使用数量	当前任务使用该材料的数量。
刀口宽度	参见 3.2。
刀口长度	参见 3.2。
最小切割尺寸	参见 3.2。
优先使用	在优化计算过程中，即使牺牲部分材料利用率也要优先使用该材料。

4.2.4 零件属性

零件属性详见下表：

属性名称	意义
编号	零件的序号。
材料类型	线材、板材、坯材三者之一。
规格	零件所用原材料的规格
名称	零件名称
X 尺寸	X 尺寸 ≥ 0

Y 尺寸	Y 尺寸 ≥ 0
Z 尺寸	Z 尺寸 ≥ 0
XY 方向	允许 X 轴向右, Y 轴向上的方向排列
YX 方向	允许 Y 轴向右, X 轴向上的方向排列
XZ 方向	允许 X 轴向右, Z 轴向上的方向排列
ZX 方向	允许 Z 轴向右, X 轴向上的方向排列
YZ 方向	允许 Y 轴向右, Z 轴向上的方向排列
ZY 方向	允许 Z 轴向右, Y 轴向上的方向排列
纹理编码	数值为 0 时表示无效。其它值表示, 具有同样纹理编码的零件必须保持同方向。
计划数量	计划切割数量。-1 表示数量无具体限制, 可根据提高材料利用率的原则自由加工。
剩余数量	执行完本切割方案后的未完成数量。
使用数量	本切割方案完成的数量。

4.2.5 改变列的位置和宽度

您可以用鼠标拖动列改变当前列所在的位置, 也可以用鼠标拖动列的分隔线改变列的宽度。

4.2.6 线材设置

原材料为线材时, X、Y、Z 三个方向的尺寸只能有一个为有效尺寸。

我们建议把 X 尺寸设为线材的长度, Y 尺寸一律设置为同一个常量 (例如原材料和零件的 Y 尺寸一律设为 100), Z 尺寸设置为 0。按照这种设置, 我们可以在作业编辑区看到和板材一样的排样效果。

4.2.7 板材设置

原材料为板材时, X、Y、Z 三个方向的尺寸只能有两个为有效尺寸。

我们建议 X、Y 设为原材料的长度和宽度 (建议水平方向的尺寸大一些, 因为切割算法是假设水平方向优先切割), Z 尺寸设置为板材的厚度, 也可以把 Z 尺寸设置为 0。

一定要注意, 原材料和零件的 Z 尺寸不一样时是无法进行切割的。

4.2.8 卷材设置

原材设置和板材类似，只是我们建议把 Y 尺寸设置为“无限制”。因为优化计算程序假设水平方向切割优先，这样才能使得优化计算程序能计算出正确结果。

4.2.9 坯材设置

原材料为坯材时，X、Y、Z 三个方向的尺寸都要根据实际情况设定。

坯材的切割实际上使用一个切割平面分割立方体。实际上有三种切割方向，按照切割平面的法线方向(也就是切割平面的垂线方向),X 方向(法线向右)、Y 方向(法线向上)、Z 方向(法线向前，即计算机屏幕的垂线方向)。参见下面图示：

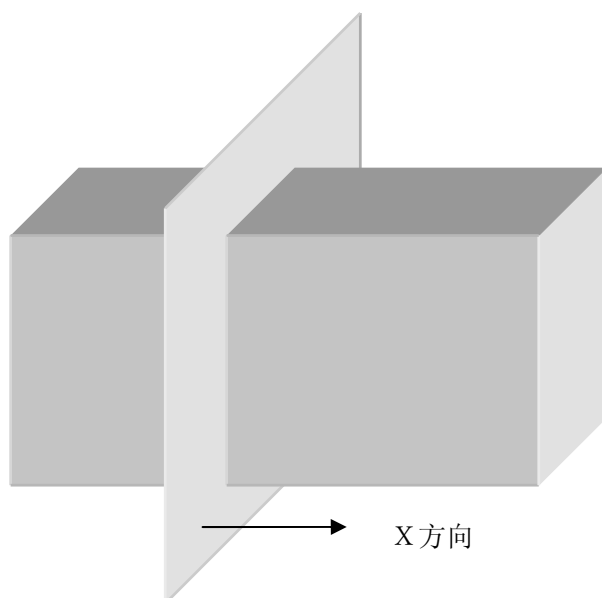


图 4.2.9.1 X 方向三维切割（极限切割 2006 支持）



图 4.2.9.2 Y 方向三维切割（极限切割 2006 支持）

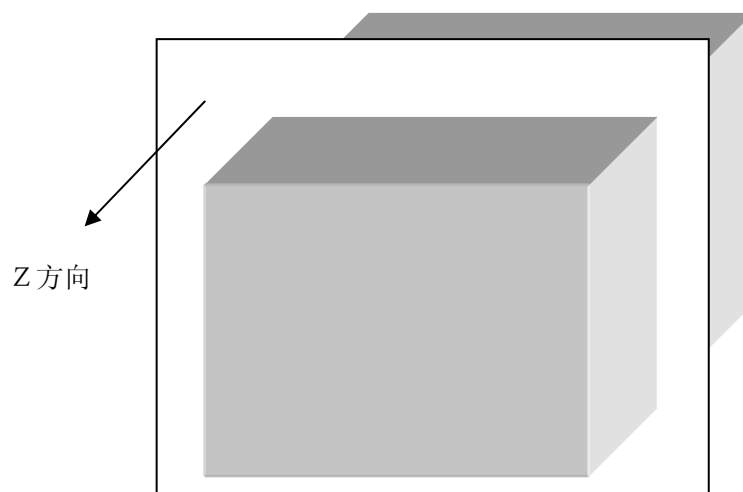


图 4.2.9.3 Z 方向三维切割（极限切割 2006 不支持）

在具体排样时，切割方向仍然只沿水平和竖直方向进行，这和板材切割方式是一样的。《极限切割 2006》不提供图 4.2.9.3 所示的切割方式。

4.3 行操作

4.3.1 添加行

如果一直接回车键或者向下移动光标键，当光标移动到表格最后一行时就会自动添加一行新的记录。

4.3.2 插入行

无论在任何一行，键入 [Insert] 键就会在当前行插入一条新记录。

4.3.3 移动行

可以直接拖动原材料或者零件的编号到希望的位置。如下图所示：



示意图 直接把编号拖放到希望的位置

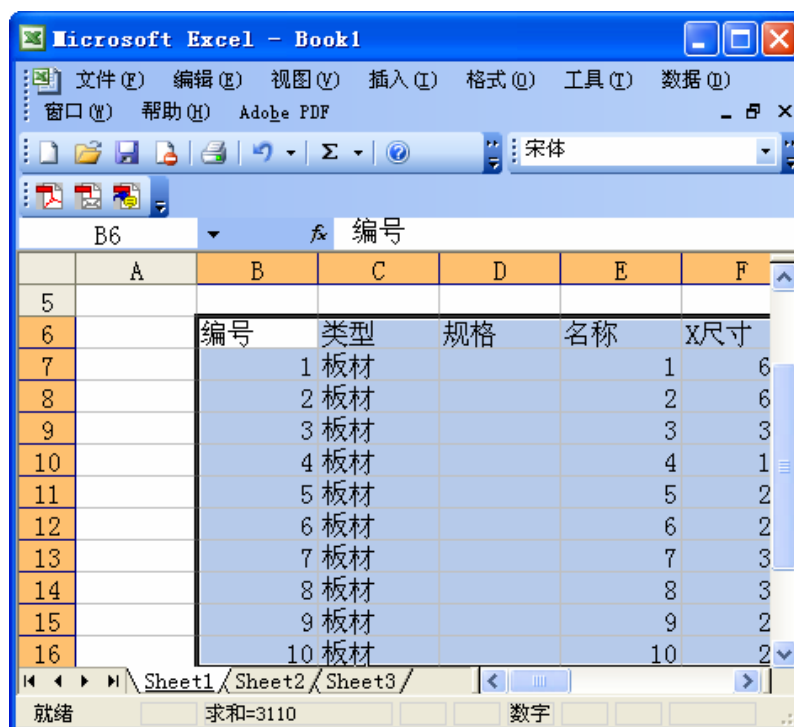
4.4 复制与粘贴

4.4.1 把数据导出到 Excel

在编辑区选择右键菜单“复制全部数据”，如下图。

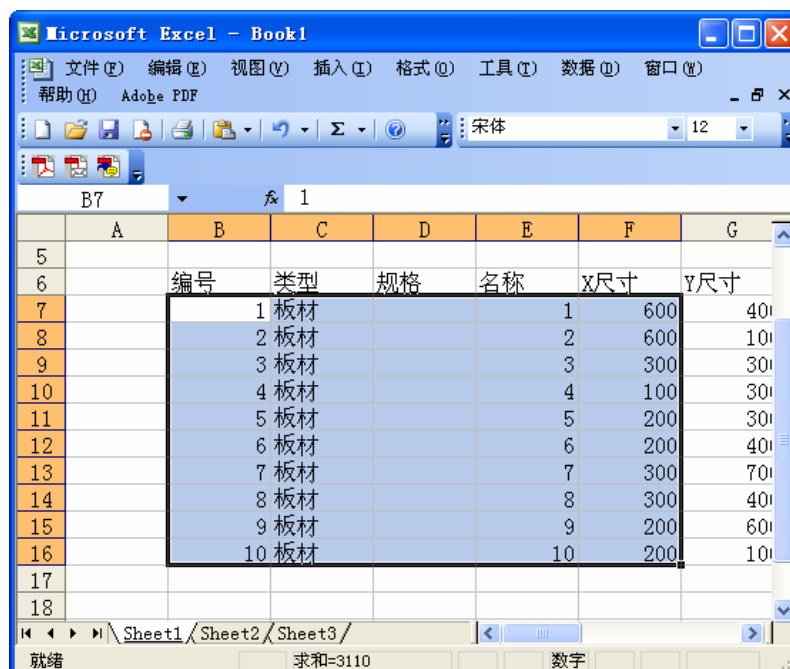


在 Excel 选择右键菜单“粘贴”，即可把编辑区数据全部复制到 Excel 中。参见下图：



4.4.2 从 Excel 导入数据

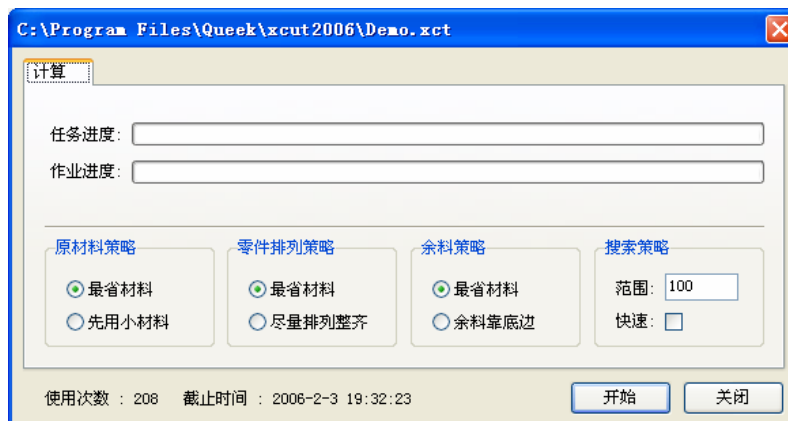
在 Excel 中选择数据，注意不要包含表头的标题。如下图：



选择右键菜单“复制”，然后回到《极限切割 2006》编辑区，选择右键菜单“粘贴”，即可把数据复制过来。

5 自动计算

选择工具栏上的“计算”按钮，即可打开计算窗口。参加下图：



5.1 设置原材料策略

- 最节省材料：目标是用最小面积的原材料完成当前切割任务。
- 先用小材料：在牺牲部分利用率的情况下，允许优先选择尺寸较小的原材料。

注意：即使在“最节省材料”策略下，《极限切割 2006》也会尽量先用小材料，除非使用较大尺寸的材料有更好的利用率。

如果希望优先使用某些材料，还可以利用原材料“优先使用”属性。详情参见 4.2.3 原材料属性。

5.2 设置零件策略

- 最节省材料：目标是用最小面积的原材料完成当前切割任务。
- 尽量排列整齐：在牺牲部分利用率的情况下，允许零件排列的更整齐一些。

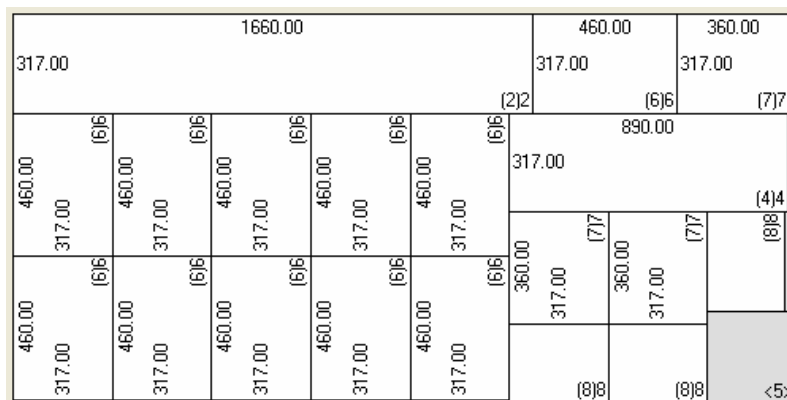


图 5.3.2 “余料靠底边”策略的计算结果

5.4 设置搜索策略

- 范围：搜索范围越大，获得最佳结果的可能性就越大，同时需要计算机的内存容量就越大。因此高速的 CPU 和大容量内存能够使《极限切割 2006》获得最佳表现。一般我们建议搜索范围设置在 1000 左右。
- 快速：如果条件不允许较长的计算时间，选择“快速”计算策略可大大加快计算速度。

注意：没有免费的午餐。高速计算总是以缩小搜索范围为代价。同样的条件，搜索速度越快，搜索结果就会越差。这几乎是普遍的规律。如果时间允许，我们总是推荐采取较大的搜索范围和“慢速”计算策略，多花几分钟计算，也许能让您省下相当可观的成本。

6 手工排样

手工排样是一项更易于使用的功能，允许你修改不适当的计算结果。

6.1 添加作业

添加新的作业有两种方式。可以直接在



6.2 删除作业

在作业标签区用右键菜单可以删除当前作业。



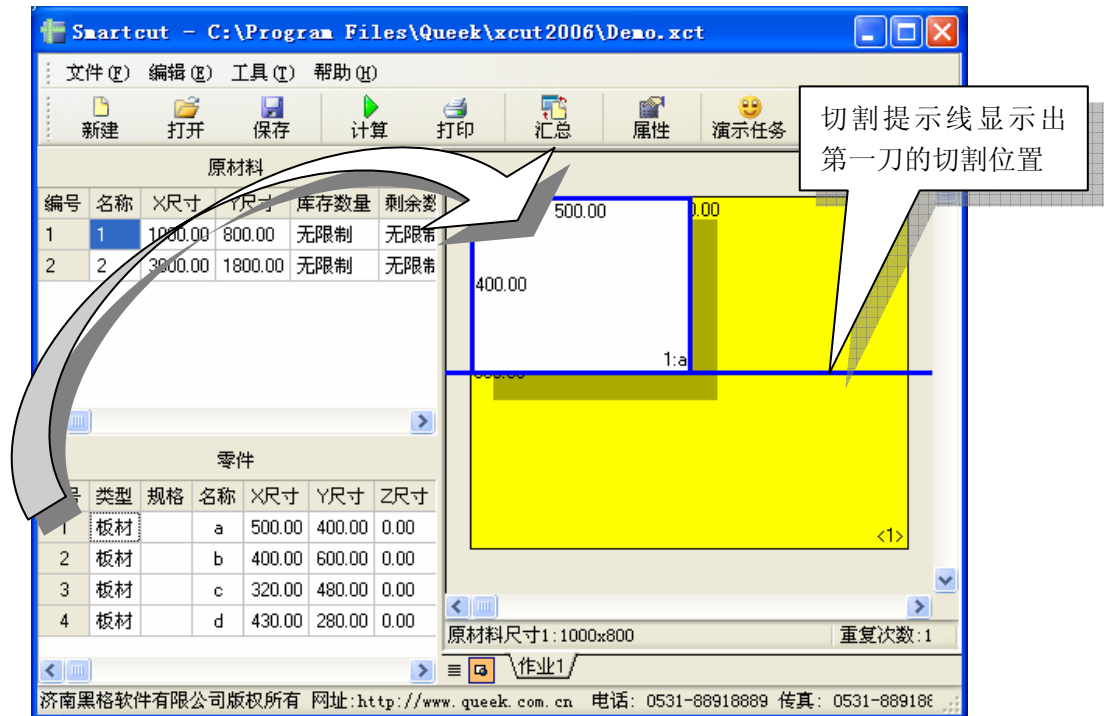
也可以选择清除全部作业，清除当前任务的全部作业。

6.3 把原材料拖放到作业区

如果当前作业还没有排放任何零件，那么可以把原材料拖入当前作业区。如果当前作业区已经有某种原材料，则会用正在拖放的原材料把原来的替换掉。

6.4 把零件拖放到作业区

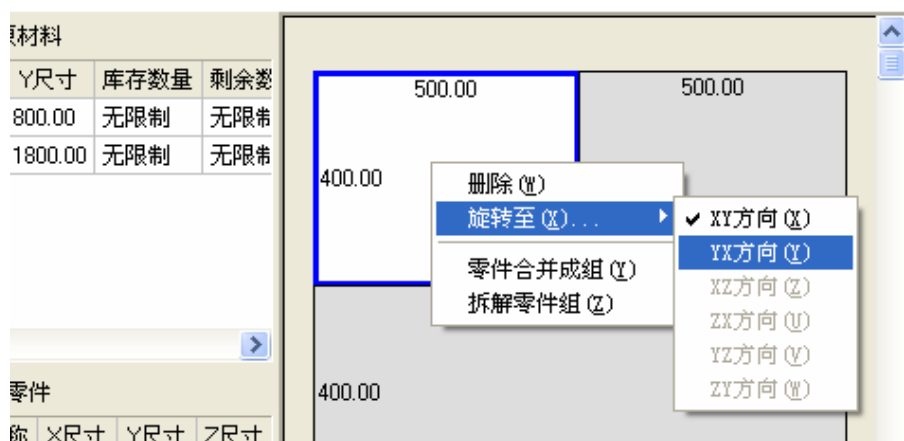
拖动零件编号到作业编辑区，可以看到如下结果。背景空白区域变成黄色，表示该零件可以放入这个区域。



有些时候，您可能发现有些零件无法放入面积比较大的空白区。这种情况下，您应该检查零件的材料类型、规格和厚度（Z 尺寸）是否与原材料一致。

6.5 转动零件

在编辑区选择右键菜单中提供的相关方向，即可实现旋转。



6.6 零件分组

如果切割任务中存在大量尺寸较小的零件，这些零件的尺寸和名称就会因零件面积较小而无法显示。

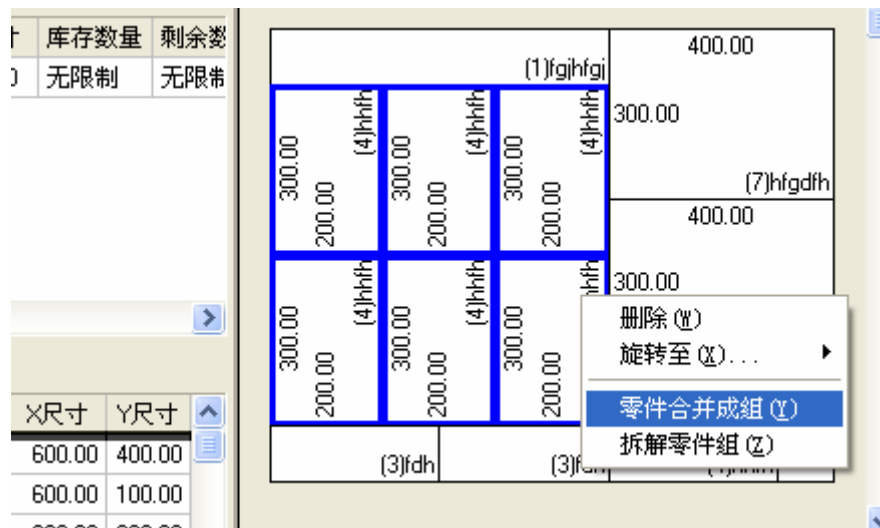


图 6.6.1 利用[Shift]+鼠标左键可以选择多个零件。通过右键菜单，可以把选择的多个零件组成一个零件组

零件分组功能可以把小零件组合成一个较大的零件组，这样分组后的作业就可以显示全部尺寸了。至于零件组内的小零件，《极限切割 2006》的打印程序可以单独打印零件组的局部图纸。

当组成零件组后，零件组处在漂浮状态。您还必须把零件组放入合适的空白区才行。

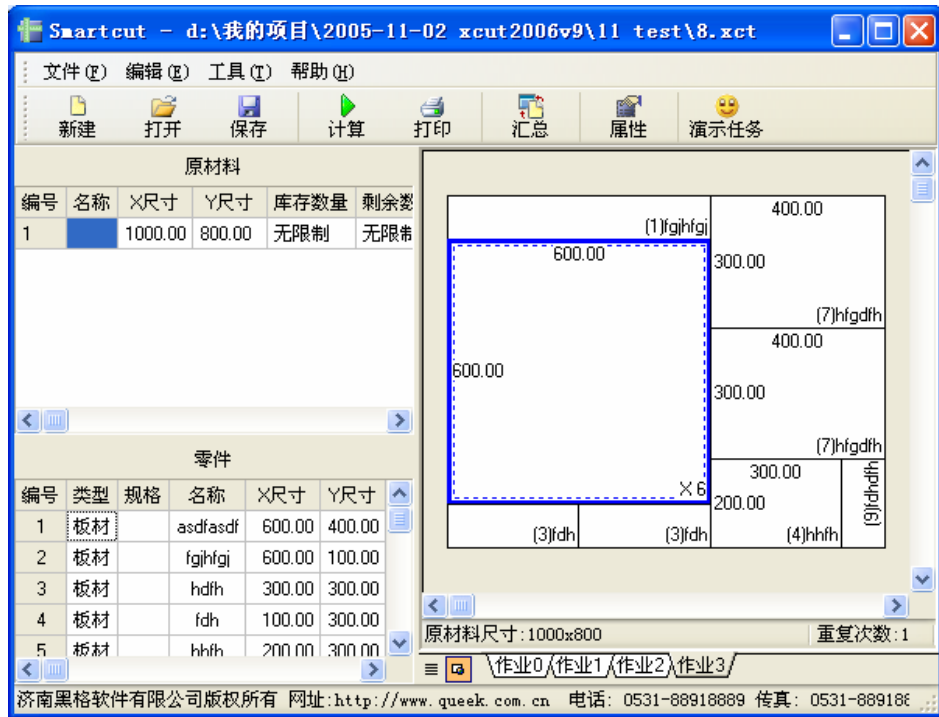
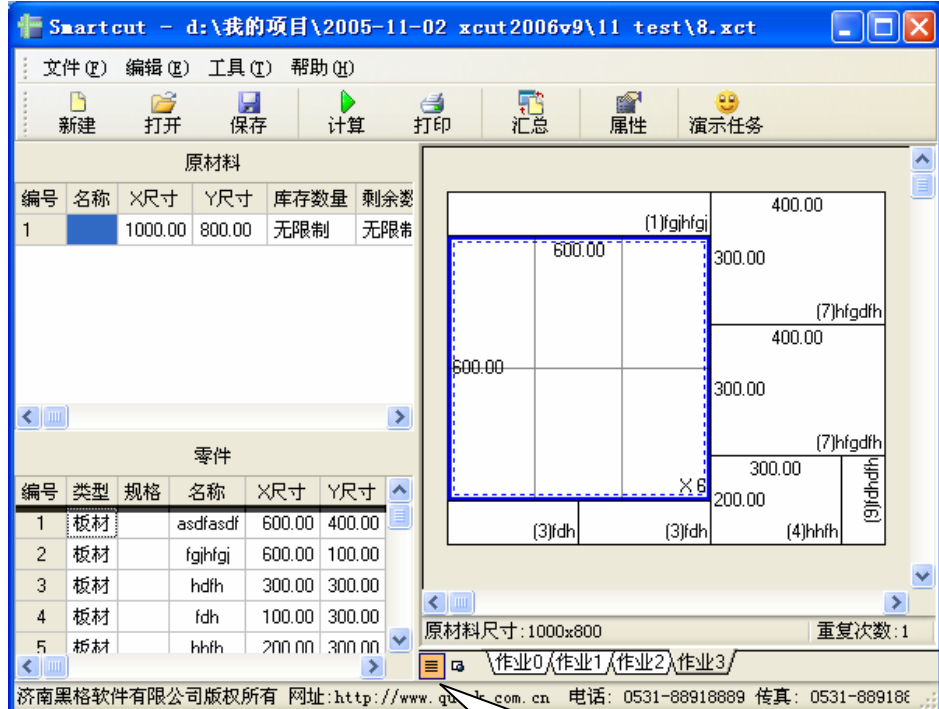


图 6.6.2 组成零件组后，用鼠标把该零件组放入空白区。

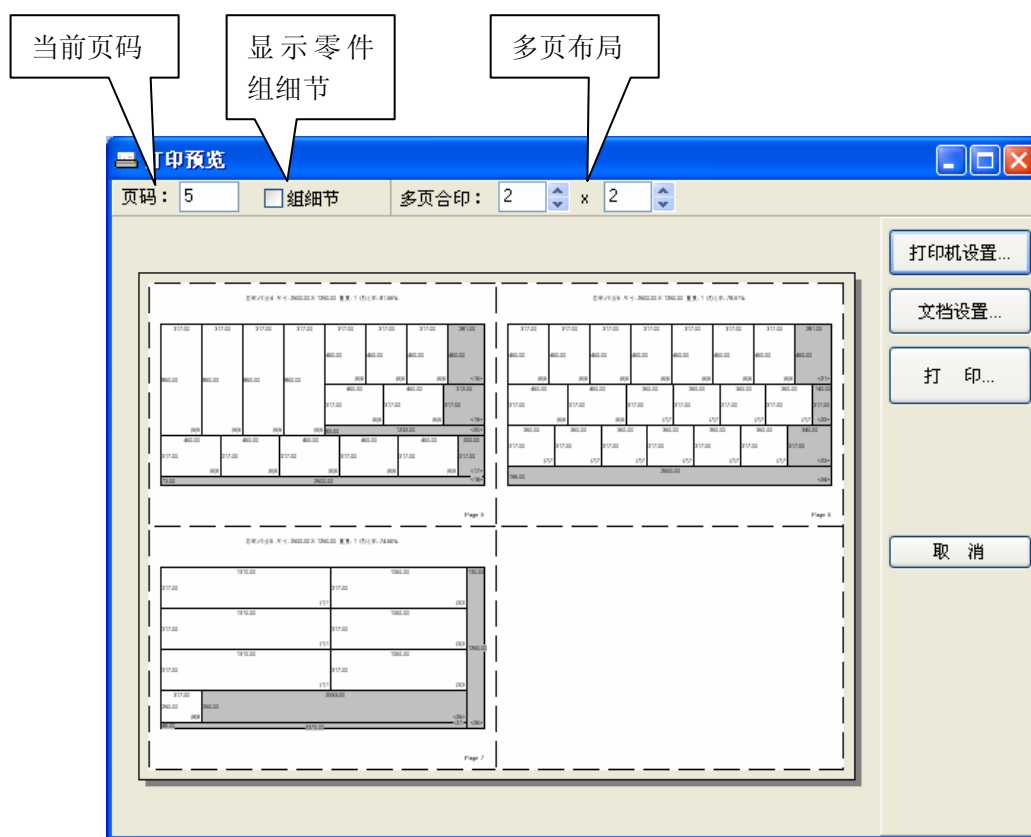


6.7 拆解零件组

选择右键单“拆解零件组”可以把零件组拆解成单个零件的形式。当拆解完成后，零件处于漂浮状态，您必须把每个零件重新放入合适的空白区域中。

7 打印

选择菜单“文件 / 打印”操作，或者直接点击工具条上的“打印”按钮，显示打印窗口。打印窗口布局如下：



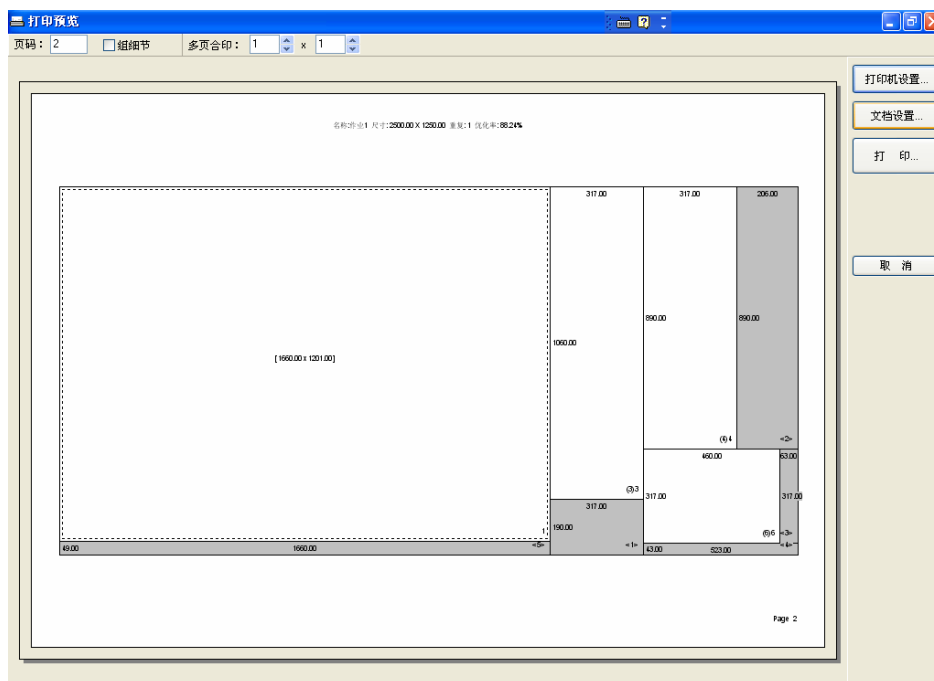
7.1 选择当前预览的内容

设置当前页码可以选择正在预览的内容，可以在打印窗口的左上角的当前页码录入筐内设置，也可以用鼠标的滚轮（有些老式鼠标可能没有滚轮）滚动设置。

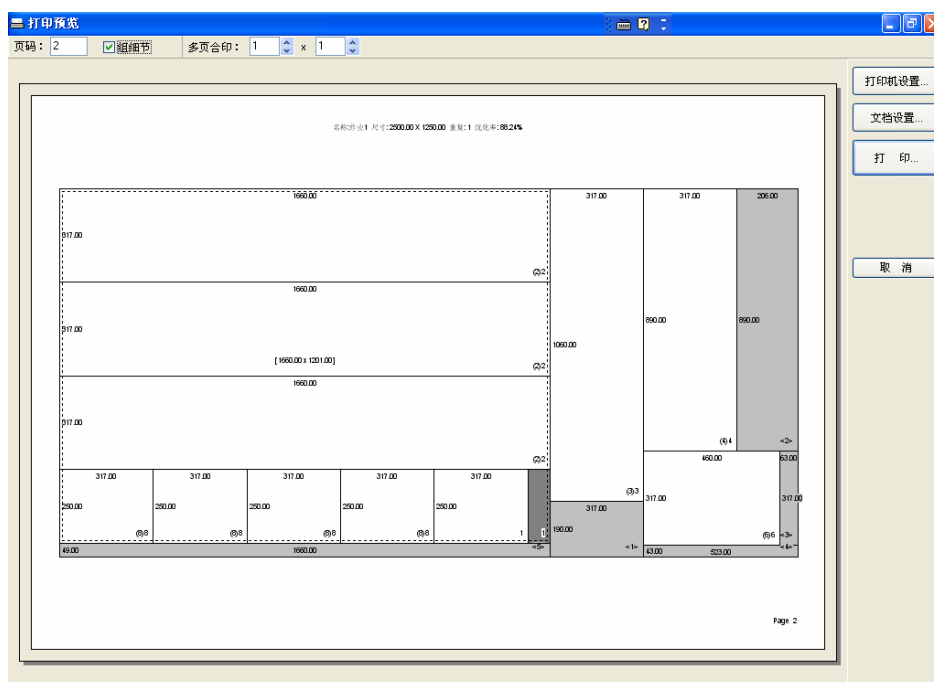
7.2 显示零件组细节

设置显示零件细节可以显示零件组内容。高级版本的打印软件能够单独打印

零件组，以便使零件组细节更清晰。
不显示细节图例：

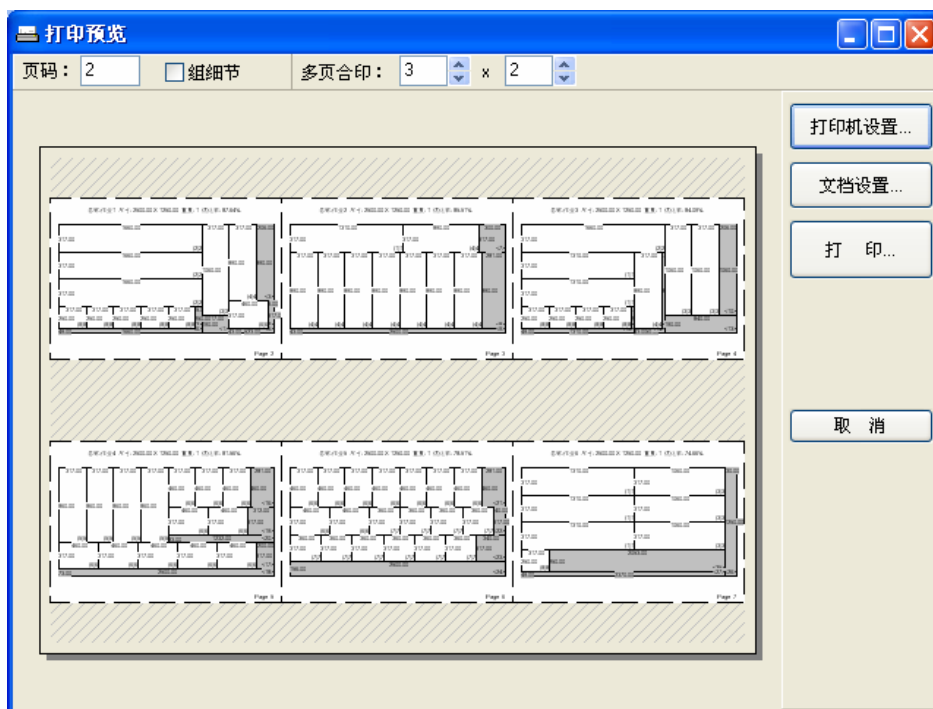


显示细节图例：



7.3 多页布局

利用多页布局，可以在一张打印纸打印多幅排样图。上图给出了 2×2 的情形，下图给出了 3×2 的情形：



7.4 打印机设置

点击“打印机设置”按钮可以显示打印机设置窗口，利用该窗口可以选择打印机型号、设置打印机属性和打印纸的尺寸和方向。



7.5 文档设置

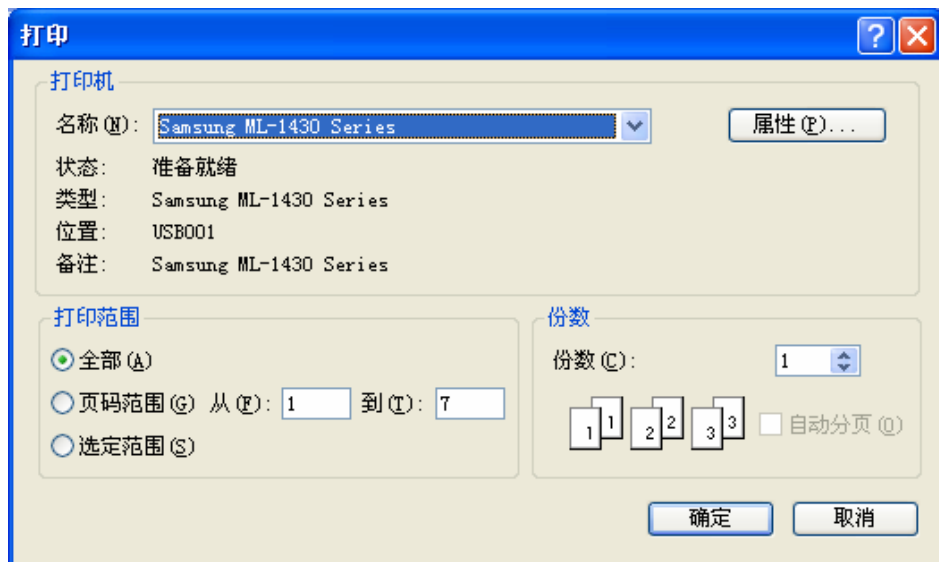
点击“文档设置”程序显示文档设置窗口，参见下图：



利用该窗口可以设置页边局、页眉、页脚、报表样式等。

7.6 打印输出

点击“打印”按钮，显示打印选择窗口。参见下图：



您可以选择页码范围，打印份数后，点击“确定”按钮，即可把要打印的内容送往打印机。

8 高级技巧

8.1 撤销与重做

《极限切割 2006》支持撤销与重做功能，如果您执行了一个错误的操作，可以通过主菜单的“编辑 / 撤销”功能，或者右键菜单的“撤销”功能恢复到错误操作之前的状态。

如果发现撤销错了，可以选择“重做”恢复到撤销操作前的状态。

8.2 使用“无限制”作为原材料数量

原材料库存数量的值为-1时表示数量没有限制。一般对于储存量较大的标准原材料可以采取这一策略。这样可以简化原材料库的管理。

更复杂的原材料库管理可以利用本系统提供的标准插件实现。

9 安装插件

《极限切割 2006》的插件功能为开拓该软件的实际应用提供了空间。除了我们为该产品提供若干标准插件和高级插件外，我们还会推出一些行业专用插件。第三方软件间开发商也会提供各类插件。这将给极限切割客户提供很多便利。

同时，插件技术也允许《极限切割 2006》和企业现有管理系统集成或者和 CIMS 设备集成。

9.1 插件类型

《极限切割 2006》提供下面十种类型的插件：

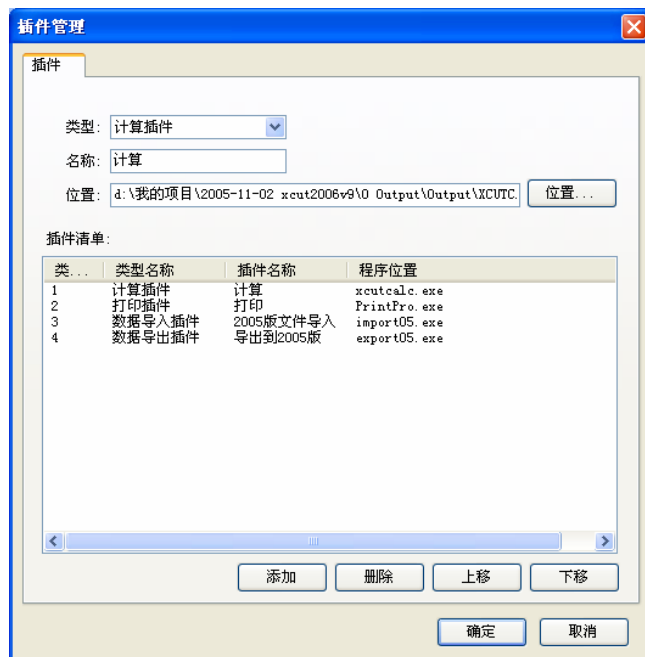
插件类型	调用方法
通用	菜单“工具”
计算	菜单“工具 / 计算”
打印	菜单“文件 / 打印”
数据导入	菜单“文件 / 导入”
数据导出	菜单“文件 / 导出”
创建文件事件响应	创建文件操作之后自动执行
打开文件事件响应	打开文件操作之后自动执行

保存文件事件响应	保存文件操作之后自动执行
打印文件事件响应	打印文件操作之后自动执行
关闭文件事件响应	关闭文件操作之后自动执行

9.2 添加插件

一般插件开发商应该提供插件安装程序。特殊情况下，可能需要用户手工安装插件。

选择菜单“工具 / 插件管理”，显示如下窗口：



选择插件类型，输入插件名称，查找插件位置，然后点击“添加”即可。

9.3 删除插件

在插件清单选择要删除的插件，然后单击“删除”即可。

9.4 标准插件

《极限切割 2006》本身带有四个标准插件：

- 计算插件
- 打印插件
- 2005 版数据导入插件
- 2005 版数据导出插件

您可以通过插件管理器查看这些插件是如何配置的。

10 开发商技术指南

《极限切割 2006》允许用户或者软件开发商开发特殊用途的插件，以扩充系统能力。本章介绍如何根据插件设计标准开发自己的插件，以及如何利用软件提供的组件加快开发速度。

10.1 系统架构

系统包括三个基本部分：

- 主程序：xcut.exe
- 编辑控件：xcutedit.ocx
- 数据组件：xcuttask.dll

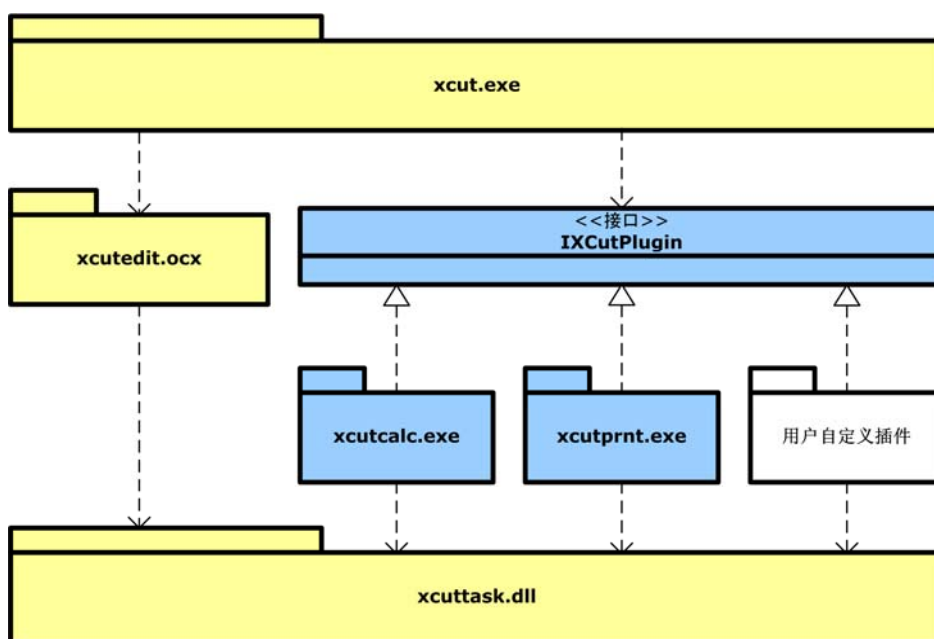


图 10.1.1 系统架构示意图

系统提供了一个插件设计标准 IXCutPlugin，上面示意图把该规范化成了接口的形式。

《极限切割 2006》自身包含两个插件：

- 优化计算程序：xcutcalc.exe
- 打印程序：xcutprnt.exe

由于这两个插件是完全按照插件标准设计的，因此，用户完全可以用自己的

算法替换掉这两个插件。同时，用户可以根据自身要求开发或者委托第三方开发相应的专用插件，这些插件可以如同我们提供的优化计算程序和打印程序一样与主系统集成在一起工作。

10.2 如何开发插件

插件的设计标准包括三个条件：

- 插件必须是一个可执行程序。一般要求扩展名为 `exe`
- 插件在被调用的时候，应该能够接受至少一个命令行参数，其最后一个参数必须是切割任务的文件名。例如我们提供的优化计算插件实际上是用这样的命令行参数被调用的：`xcutcalc test.xct`，该命令行参数的执行结果是，`xcutcalc.exe` 从文件 `test.xct` 中读取任务数据，并把计算结果写回该文件。
- 如果插件对任务文件提供了修改，插件必须把结果写回命令行参数提供的任务文件中。

安装光盘提供的插件的实际例子，其中包括 `import05.exe` 和 `export05.exe` 的源代码。

10.3 *.xct 文件格式

任务文件被保存成 `*.xct` 格式，实际上 `*.xct` 格式是标准的 `INI` 文件，可以用 `Windows API` 提供的 `INI` 文件访问函数访问。

例如，在 `Delphi` 中，可以用 `TIniFile` 类访问此类文件。下面给出该文件的一个实例，并给出相关的解释。

```
[Root]//任务根属性
Display Mode=0 //显示模式，决定是否显示零件组细节
Show Size=1 //是否显示零件尺寸
Show Text=1 //是否显示零件名称
Size Dec=2 //小数点位数
Raws Count=1 //原材料清单的行数
Parts Count=8 //零件清单的行数
Jobs Count=6 //作业的页数
Fields Count=0 //保留
Default Material Type=1 //详见属性窗口。默认材料类型 0=线材 1=板材 2=坯材
Default Cut Width=0 //默认切割宽度
Default Cut Length=-1 //默认切割长度
Default Min Cut Size=0 //最小切割尺寸
Unit Rate Length=1000000
```

```
Unit Rate Area=1
Length Unit=毫米
Area Unit=平方米
[Raws]//原材料清单
Raw[0].MaterType=1
Raw[0].MaterSpec=
Raw[0].MaterName=2
Raw[0].XSize=103000
Raw[0].YSize=125000
Raw[0].ZSize=0
Raw[0].CutWidth=0
Raw[0].CutLength=-1
Raw[0].MinCutSize=0
Raw[0].StockNumber=-1
Raw[0].UsedNumber=33
Raw[0].UseFirst=0
[Parts]//零件清单
Part[0].MaterType=1
Part[0].MaterSpec=
Part[0].MaterName=B
Part[0].XSize=33000
Part[0].YSize=64000
Part[0].ZSize=0
Part[0].DirId=0
Part[0].DirXY=1
Part[0].DirYX=1
Part[0].DirXZ=0
Part[0].DirZX=0
Part[0].DirYZ=0
Part[0].DirZY=0
Part[0].PlanNumber=10
Part[0].UsedNumber=10
Part[1].MaterType=1
Part[1].MaterSpec=
Part[1].MaterName=C
Part[1].XSize=59500
Part[1].YSize=81300
Part[1].ZSize=0
Part[1].DirId=0
Part[1].DirXY=1
Part[1].DirYX=1
Part[1].DirXZ=0
Part[1].DirZX=0
Part[1].DirYZ=0
```

```

Part[1].DirZY=0
... ..
Part[7].PlanNumber=10
Part[7].UsedNumber=10
[Jobs]//作业清单
Job[0].Text=作业 1
Job[0].RowIndex=0
Job[0].Width=103000
Job[0].Height=125000
Job[0].ViewX=50
Job[0].ViewY=50
Job[0].RepeatNumber=10
Job[0].CutCount=2 //零件数量
Job[0].SpaceCount=0 //余料数量
Cut[0,0].Text=(3)D //排列在作业上的零件
Cut[0,0].PartIndex=2
Cut[0,0].PartDir=0
Cut[0,0].Left=0
Cut[0,0].Top=0
Cut[0,0].Width=81000
... ..
Cut[5,1].GroupType=0
Cut[5,1].SelectState=0
Cut[5,1].DragState=0

```

上述格式仅供参考，随着产品的升级换代，有可能根据需要变动。

10.4 xcuttask.dll

声明：未经济南黑格软件有限公司许可，不得将此组件用于商业目的。

xcuttask.dll 是一个 Windows 下的 COM 数据组件（也称为类库），它用于简化编制插件的数据访问工作。利用该组件，可以存取*.xct 数据文件。

这里介绍该组件所包含的几个主要的 COM 类，语法采用 Object Pascal。

ITaskHistory: 切割任务的修改历史纪录。这是实现编辑操作中 Undo、Redo 的关键。

```

//清除历史纪录。
procedure Clear;

//添加一个历史纪录。在执行一个需要 Undo 的操作前调用该方法。

```

```

procedure Add;

//撤销最后一次的操作。
procedure Undo;

//重做刚才被撤销的操作。
procedure Redo;

//直接转到某历史纪录中去。
procedure GotoTask(Index: Integer);

//当前正在使用的任务。
property CurTask: ITask;

//是否允许撤销。
property CanUndo: WordBool;

//是否允许重做。
property CanRedo: WordBool;

//当前任务在操作历史纪录中的顺序号。
property TaskIndex: Integer;

```

ITask: 切割任务——用于保存一个完整的切割任务。

```

//从另一个任务复制数据。
procedure Copy(const Value: ITask);

//注意，在切割文件中，原材料和零部件等尺寸一律用整数表示。但是在用户界面里，系统允许输入浮点数。因此，ITask 提供了证书和浮点数的转换算法。
function IntToFloat(Param: Integer): Double; //把整数转换为浮点数
function FloatToInt(Param: Double): Integer; //把浮点数转换成整数

//数据读盘与存盘。利用这两个方法可以存取切割文件。
procedure SaveToFile(const FileName: WideString);
procedure LoadFromFile(const FileName: WideString);

//清除全部数据。
procedure Clear;

//更新数据。因为有些属性是根据排样情况推算出来的，如果有必要调用下面方法是这个数据重新计算这些属性。
procedure RefreshData;

//重新设定作业中的图形名称。这包括：作业名称、零件名称和余料名称。

```

```
procedure SetJobNames;  
  
//原材料清单、零部件清单和排样作业清单  
property Rows: IRawCollection;  
property Parts: IPartCollection;  
property Jobs: IJobCollection;  
  
property SizeDec: Integer;  
property ShowSize: WordBool;  
property SizeRate: Integer;  
property ShowText: WordBool;  
property DisplayMode: DisplayMode;  
property DefaultMaterType: MaterType;  
property DefaultCutWidth: Integer;  
property DefaultCutLength: Integer;  
property DefaultMinCutSize: Integer;  
property UnitRateLength: Integer;  
property UnitRateArea: Integer;  
property LengthUnit: WideString;  
property AreaUnit: WideString;  
  
//下面这些数据是根据排样数据计算而来，保留这些属性是为了提高应用系统速度。  
RefreshData 方法会重新计算这些属性。  
property GoodValue: Double;  
property TotalRawCount: Integer;  
property TotalRawArea: Double;  
property TotalPartCount: Integer;  
property TotalPartArea: Double;
```

IRaw——原材料

```
procedure Copy(const Value: IRaw);  
property MaterType: MaterType;  
property MaterSpec: WideString;  
property MaterName: WideString;  
property XSize: Integer;  
property YSize: Integer;  
property ZSize: Integer;  
property StockNumber: Integer;  
property LeftNumber: Integer;  
property CutWidth: Integer;  
property CutLength: Integer;  
property MinCutSize: Integer;  
property UsedNumber: Integer;  
property UseFirst: WordBool;
```

IRawCollection——原材料清单

```
//添加原材料, 返回新添加的原材料
function Add: IRaw;

//删除第 Index 行原材料
procedure Delete(Index: Integer);

//在第 Index 行插入原材料
function Insert(Index: Integer): IRaw;

//把 CurIndex 行移动到 NewIndex 行
procedure Move(CurIndex: Integer; NewIndex: Integer);

//复制原材料清单
procedure Copy(const Value: IRawCollection);

//清除全部内容
procedure Clear;

//原材料清单行数
property Count: Integer;

//获取第 Index 个原材料数据
property Items[Index: Integer]: IRaw;
```

IPart——零件

```
property MaterType: MaterType;
property MaterSpec: WideString;
property MaterName: WideString;
property XSize: Integer;
property YSize: Integer;
property ZSize: Integer;
property DirId: Integer;
property DirXY: WordBool;
property DirYX: WordBool;
property DirXZ: WordBool;
property DirZX: WordBool;
property DirYZ: WordBool;
property DirZY: WordBool;
property PlanNumber: Integer;
property LeftNumber: Integer;
property UsedNumber: Integer;
property Product: WideString;
```

IPartCollection

```
function Add: IPart; safecall;
procedure Delete(Index: Integer); safecall;
function Insert(Index: Integer): IPart; safecall;
procedure Move(CurIndex: Integer; NewIndex: Integer); safecall;
procedure Copy(const Value: IPartCollection); safecall;
procedure Clear; safecall;
property Count: Integer read Get_Count;
property Items[Index: Integer]: IPart read Get_Items;
```

IJob——作业

```
procedure Copy(const Value: IJob);

//原材料行数
property RawIndex: Integer;

//名称
property Text: WideString;

//零件清单
property Cuts: ICutCollection;

//余料清单
property Spaces: ISpaceCollection;

//视图中心坐标和方缩比例
property ViewX: Integer;
property ViewY: Integer;
property Zoom: Double;

//重复执行次数
property RepeatNumber: Integer;

//当前选区
property SelectedLeft: Integer;
property SelectedTop: Integer;
property SelectedWidth: Integer;
property SelectedHeight: Integer;

//当前作业尺寸。由于原材料可能尺寸没有限制（-1，例如卷材），作业的尺寸和原材料尺寸可能不一致。
property Width: Integer;
property Height: Integer;
```



```
//切割利用率
property GoodValue: Double;
```

IJobCollection

```
function Get_Count: Integer;
function Get_Items(Index: Integer): IJob;
function Add: IJob;
procedure Delete(Index: Integer);
function Insert(Index: Integer): IJob;
procedure Move(CurIndex: Integer; NewIndex: Integer);
procedure Copy(const Value: IJobCollection);
procedure Clear;
property Count: Integer;
property Items[Index: Integer]: IJob;
```

ICut——排样图中的零件

```
procedure Copy(const Value: ICut);

//零件行号、放置方向和位置大小
property PartIndex: Integer;
property PartDir: MaterDir;
property Left: Integer;
property Top: Integer;
property Width: Integer;
property Height: Integer;

//零件组首个零件的序号
property GroupCutIndex: Integer;

//组类型: 普通组(cgtGroup)、同一零件的排列(cgtArray)、组内子成员(cgtChild)、
非组成员(cgtNone)
property GroupType: CutGroupType;

//选择状态与拖放状态
property SelectState: CutSelectState;
property DragState: CutDragState;

//备份数据: 当零件分组时, 需要把分组前的信息备份, 以便拆分零件组时恢复单个零件
信息
property LeftCopy: Integer;
property TopCopy: Integer;
property WidthCopy: Integer;
property HeightCopy: Integer;
```

```
property Text: WideString;  
property TextCopy: WideString;
```

ICutCollection

```
function Add: ICut;  
procedure Delete(Index: Integer);  
procedure Copy(const Value: ICutCollection);  
procedure Clear;  
property Count: Integer;  
property Items[Index: Integer]: ICut;
```

ISpace——切割余料

```
procedure Copy(const Value: ISpace);  
property Left: Integer;  
property Top: Integer;  
property Width: Integer;  
property Height: Integer;  
property Text: WideString;
```

ISpaceCollectio

```
function Add: ISpace;  
procedure Delete(Index: Integer);  
procedure Copy(const Value: ISpaceCollection);  
procedure Clear;  
property Count: Integer;  
property Items[Index: Integer]: ISpace;
```

10.5 xcutedit.ocx

声明：未经济南黑格软件有限公司许可，不得将此组件用于商业目的。

ITaskEdit——任务编辑器

```
//刷新显示  
procedure Refresh;  
  
//显示属性页  
procedure ShowPropertyPage;  
  
//显示汇总页  
procedure ShowSummary;
```

```
//任务历史纪录
property TaskHistory: ITaskHistory;

//放缩方式
property ZoomStyle: ZoomStyle;

//下面是 ActiveX 控件通用属性
property Visible: WordBool;
property AutoScroll: WordBool;
property AutoSize: WordBool;
property AxBorderStyle: TxActiveFormBorderStyle;
property Caption: WideString;
property Color: OLE_COLOR;
property Font: IFontDisp;
property KeyPreview: WordBool;
property PixelsPerInch: Integer;
property PrintScale: TxPrintScale;
property Scaled: WordBool;
property Active: WordBool;
property DropTarget: WordBool;
property HelpFile: WideString;
property ScreenSnap: WordBool;
property SnapBuffer: Integer;
property DoubleBuffered: WordBool;
property AlignDisabled: WordBool;
property VisibleDockClientCount: Integer;
property Enabled: WordBool;

//事件响应函数
//数据发生改变
procedure OnDataChange;

//ActiveX 通用事件响应函数
procedure OnActivate;
procedure OnClick;
procedure OnCreate;
procedure OnDblClick;
procedure OnDestroy;
procedure OnDeactivate;
procedure OnKeyPress(var Key: Smallint);
procedure OnPaint;
```

10.6 xcutcalc.exe

声明：未经济南黑格软件有限公司许可，不得将此组件用于商业目的。

xcutcalc.exe 是系统提供的标准计算插件，该插件通过命令行参数提供的文件名读取任务，然后计算排样结果，并把结果保存到任务文件。

您可以开发自己的计算插件，把该插件替换掉。

10.7 xcutprnt.exe

声明：未经济南黑格软件有限公司许可，不得将此组件用于商业目的。

xcutprnt.exe 是系统提供的标准打印插件，该插件通过命令行参数提供的文件名读取任务，然后打印排样结果。

您可以开发自己的打印插件，把该插件替换掉。

10.8 import05.exe

声明：本组件为免费组件。

import05.exe 可以导入《极限切割 2005》的数据文件。

您可以开发自己的数据导入插件，导入其它格式的数据文件。

10.9 export05.exe

声明：本组件为免费组件。

export05.exe 可以把当前任务导出到《极限切割 2005》格式。

您可以开发自己的数据导出插件，导出其它格式的数据文件。

10.10 import05.exe 和 export05 的主要算法源代码

```
unit Unit2005;  
  
interface
```

```
function Get2005FileName(FileName2006 : string) : string;
procedure From2005To2006(FileName2005, FileName2006 : string);
procedure From2006To2005(FileName2006, FileName2005 : string);

implementation

uses xcuttask_TLB, SysUtils, Forms;

var MaxProgressValue, ProgressValue : Integer;

function Get2005FileName(FileName2006 : string) : string;
begin
  Result := Copy(FileName2006, 1, Length(FileName2006) - 3) + 'cut';
end;

procedure From2005To2006(FileName2005, FileName2006 : string);
var
  Task : ITask;
  Raw : IRaw;
  Part : IPart;
  Job : IJob;
  Cut : ICut;
  Space : ISpace;

  RawCount, PartCount, JobCount, CutCount, SpaceCount,
  CutWidth, CutLength, MinCutSize : Integer;
  SizeRate : Integer;

  f : TextFile;
  i, j, n : longint;
  s : string;
  version : integer;
  d : double;
begin
  Task := CoTask.Create;
  AssignFile(f, FileName2005);
  Reset(f);
  try
    SizeRate := 100;
    Task.SizeDec := 2;
    Readln(f, s);
    if (s = 'Version=3') then
```

```
begin
  Readln(f, s);
  RawCount := StrToInt(s);
  Readln(f, s);
  PartCount := StrToInt(s);
  Readln(f, s);
  JobCount := StrToInt(s);
  Readln(f, d);
  CutWidth := Round(d * SizeRate);
  Readln(f, d);
  CutLength := Round(d * 100);
  try
    Readln(f, n);
  except
    n := 0;
  end;
  Readln(f, n);
  MaxProgressValue := RawCount + PartCount + JobCount;
  ProgressValue := 0;
  for i:= 0 to RawCount - 1 do
  begin
    Raw := Task.Raws.Add;
    Readln(f, s); Raw.MaterName := s;
    Readln(f, d); Raw.XSize := Round(d * SizeRate);
    Readln(f, d); Raw.YSize := Round(d * SizeRate);
    Readln(f, n); Raw.StockNumber := n;
    Readln(f, n);
  end;
  for i := 0 to PartCount - 1 do
  begin
    Part := Task.Parts.Add;
    Readln(f, s); Part.MaterName := s;
    Readln(f, d); Part.XSize := Round(d * SizeRate);
    Readln(f, d); Part.YSize := Round(d * SizeRate);
    Readln(f, n);
    Readln(f, n); Part.PlanNumber := n;
    Readln(f, n);
    Part.DirXY := True;
    Part.DirYX := Boolean(n);
  end;
  for i := 0 to JobCount - 1 do
  begin
    Job := Task.Jobs.Add;
    Readln(f, n); Job.RawIndex := n;
```

```
Job.Text := '作业' + IntToStr(i + 1);
Job.Width := Task.Raws.Items[Job.RawIndex].XSize;
Job.Height := Task.Raws.Items[Job.RawIndex].YSize;
Readln(f, n); Job.RepeatNumber := n;
Readln(f, s);
for j := 1 to StrToInt(s) do
begin
  Readln(f, s);
  Readln(f, s);
  Readln(f, s);
end;
Readln(f, s);
CutCount := StrToInt(s);
for j := 0 to CutCount - 1 do
begin
  Cut := Job.Cuts.Add;
  Readln(f, n); Cut.PartIndex := n;
  Readln(f, d); Cut.Left := Round(d * SizeRate);
  Readln(f, d); Cut.Top := Round(d * SizeRate);
  Readln(f, n);
  if not Boolean(n) then
  begin
    Cut.Width := Task.Parts.Items[Cut.PartIndex].XSize;
    Cut.Height := Task.Parts.Items[Cut.PartIndex].YSize;
  end
  else
  begin
    Cut.Width := Task.Parts.Items[Cut.PartIndex].YSize;
    Cut.Height := Task.Parts.Items[Cut.PartIndex].XSize;
  end;
end;
Readln(f, s);
SpaceCount := StrToInt(s);
for j := 0 to SpaceCount - 1 do
begin
  Space := Job.Spaces.Add;
  Readln(f, s); Space.Text := s;
  Readln(f, d); Space.Left := Round(d * SizeRate);
  Readln(f, d); Space.Top := Round(d * SizeRate);
  Readln(f, d); Space.Width := Round(d * SizeRate);
  Readln(f, d); Space.Height := Round(d * SizeRate);
  Readln(f, n); //Plan.Jobs[i].SpcRects[j].Count);
end;
end;
```

```
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s);
Readln(f, s); Task.LengthUnit := s;
Readln(f, s); Task.AreaUnit := s;
Readln(f, d); Task.UnitRateLength := Round(d);
Readln(f, d); Task.UnitRateArea := Round(d);
Readln(f, n);
Readln(f, n);
for i := 0 to n - 1 do
begin
  Readln(f, s);
  Readln(f, s);
  Readln(f, s);
  Readln(f, s);
end;
end;
Readln(f, s);
Readln(f, d);
Readln(f, s);
Readln(f, s);
Readln(f, n);
Readln(f, s);
Readln(f, n);
for i := 0 to n - 1 do
begin
  Readln(f, n);
end;
Readln(f, s);
Readln(f, s);
```



```
    Readln(f, s);
    Readln(f, s);
    Readln(f, s);
finally
    CloseFile(f);
end;
Task.RefreshData;
Task.SetJobNames;
Task.SaveToFile(ParamStr(1));
end;

procedure From2006To2005(FileName2006, FileName2005 : string);
var
    Task : ITask;
    Raw : IRaw;
    Part : IPart;
    Job : IJob;
    Cut : ICut;
    Space : ISpace;

    RawCount, PartCount, JobCount, CutCount, SpaceCount,
    CutWidth, CutLength, MinCutSize : Integer;
    SizeRate : Integer;

    f : TextFile;
    i, j : integer;
begin
    Task := CoTask.Create;
    Task.LoadFromFile(FileName2006);

    MaxProgressValue := Task.Raws.Count + Task.Parts.Count +
Task.Jobs.Count;
    ProgressValue := 0;

    AssignFile(f, FileName2005);
    Rewrite(f);
    try
        Writeln(f, 'Version=3');
        Writeln(f, Task.Raws.Count);
        Writeln(f, Task.Parts.Count);
        Writeln(f, Task.Jobs.Count);
        Writeln(f, Task.Raws.Items[0].CutWidth);
        Writeln(f, Task.Raws.Items[0].CutLength);
        Writeln(f, 0); //Ord(Plan.CalcStyle));
```

```
Writeln(f, 0); //Plan.SearchTime);
for i:= 0 to Task.Raws.Count - 1 do
begin
  Raw := Task.Raws.Items[i];
  Writeln(f, Raw.MaterName);
  Writeln(f, Raw.XSize / 100);
  Writeln(f, Raw.YSize / 100);
  Writeln(f, Raw.LeftNumber);
  Writeln(f, Raw.StockNumber);
end;
for i := 0 to Task.Parts.Count - 1 do
begin
  Part := Task.Parts.Items[i];
  Writeln(f, Part.MaterName);
  Writeln(f, Part.XSize / 100);
  Writeln(f, Part.YSize / 100);
  Writeln(f, Part.LeftNumber);
  Writeln(f, Part.PlanNumber);
  Writeln(f, Ord(Part.DirYX));
end;
for i := 0 to Task.Jobs.Count - 1 do
begin
  Job := Task.Jobs.Items[i];
  Writeln(f, Job.RawIndex);
  Writeln(f, Job.RepeatNumber);
  Writeln(f, 0);
  Writeln(f, Job.Cuts.Count);
  for j := 0 to Job.Cuts.Count - 1 do
  begin
    Cut := Job.Cuts.Items[j];
    Part := Task.Parts.Items[Cut.PartIndex];
    Writeln(f, Cut.PartIndex);
    Writeln(f, Cut.Left / 100);
    Writeln(f, Cut.Top / 100);
    Writeln(f, ord((Cut.Width = Part.YSize) and (Cut.Height =
Part.XSize)));
  end;
  Writeln(f, Job.Spaces.Count);
  for j := 0 to Job.Spaces.Count - 1 do
  begin
    Space := Job.Spaces.Items[j];
    Writeln(f, Space.Text);
    Writeln(f, Space.Left / 100);
    Writeln(f, Space.Top / 100);
```

```
    Writeln(f, Space.Width / 100);
    Writeln(f, Space.Height / 100);
    Writeln(f, 1)//Space.Plan.Jobs[i].SpcRects[j].Count);
  end;
end;
Writeln(f, FileName2006);
Writeln(f, Now);
Writeln(f, '');
Writeln(f, '');
Writeln(f, '');
Writeln(f, '');
Writeln(f, '');
Writeln(f, '0');
Writeln(f, 1);
Writeln(f, 0);
Writeln(f, $FFFFFF);
Writeln(f, 1);
Writeln(f, 0);
Writeln(f, $FFFFFF);
Writeln(f, 1);
Writeln(f, 0);
Writeln(f, $CCCCCC);
Writeln(f, Task.LengthUnit);
Writeln(f, Task.AreaUnit);
Writeln(f, Task.UnitRateLength);
Writeln(f, Task.UnitRateArea);
Writeln(f, 0);
Writeln(f, 0);
Writeln(f, '');
Writeln(f, Task.Raws.Items[0].MinCutSize);
Writeln(f, False);
Writeln(f, False);
Writeln(f, 0);
Writeln(f, 100);
Writeln(f, Task.Parts.Count);
for i := 0 to Task.Parts.Count - 1 do
begin
  Writeln(f, 0);
end;
Writeln(f, False);
Writeln(f, True);
Writeln(f, True);
Writeln(f, 0);
Writeln(f, 0);
```

```
finally
  CloseFile(f);
end;
end;
end.
```

11 售后服务和技术支持

11.1 产品更新

产品更新请登陆网站 <http://www.queek.com.cn>，查找本产品升级版本和相关资料。

11.2 联系方式

使用过程中遇到的技术困难，或者对产品改进提出建设性意见，可以与济南黑格软件有限公司客户服务中心联系。

济南黑格软件有限公司客户服务中心联系方式：

电话：0531-88918889 传真：88918886

电子邮件：kefu@queek.com.cn