

TRIBON M3 PLATE NESTING

(本说明为增强版，以后慢慢改进，看不懂请问相关高手，别找我，哈哈)

菜☆鸟☆张 2007-01-24

以下为常用快捷键说明：(这些快捷键是系统管理人员根据工作者要求来创建的，个别工具条为 Tribon 自带，每个公司的快捷键都不怎么一样，希望各位能把下面的工具条路径记下来，以方便以后工作，^_^)



1: 新建立一张套料版图

路径: Nesting → Nest job → New

2: 打开旧的套料图纸

路径: Nesting → Nest job → Open

3: 保存现在打开的套料图纸

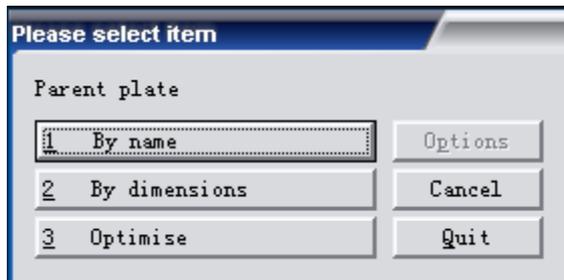
路径: Nesting → Nest job → Save

4: 图纸另存为

路径: Nesting → Nest job → Rename

5: 改变钢板的规格，材质，和是否对称套料（对称与不对称变换方法请来问，比较复杂）

路径: Nesting → Nesting parts → Exchange → Rawplate



改变钢板大小选择 2: By dimensions 系统会弹出下面的对话框:

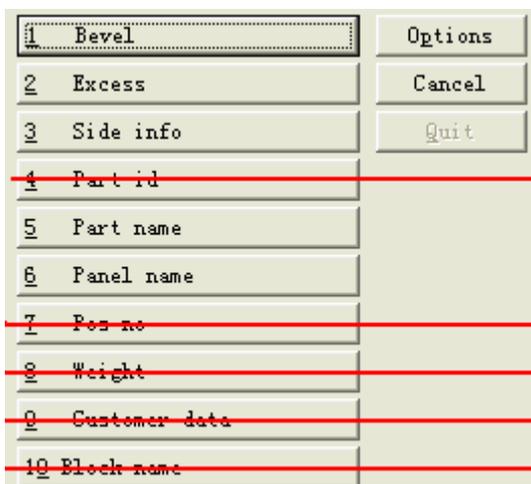


注：一定要根据订货清单或船厂提供的钢板规格进行修改

- 6: 移动零件
路径: Nesting → Nesting parts → Transform
- 7: 更换零件
路径: Nesting → Nesting parts → Move part
- 8: 删除过桥
路径: Nesting → Delete → Bridge
- 9: 删除起割点
路径: Nesting → Delete → Start
- 10: 删除零件
路径: Nesting → Delete → Part
- 11: 删除草图 (牵涉到的内容比较多, 不懂的请来问)
路径: Nesting → Delete → Sketch



- 1: 启动虚拟切割机
路径: (我也没找到再哪, 点这个按钮就行, 嘿嘿.....)
- 2: 套料版图与图面之间的转换
路径: Nesting → Shop dwg I/O → Display burning sketch
- 3: 自动添加起割点
路径: Nesting → Nesting tools → Start → Automatic
- 4: 增加零件
路径: Nesting → Nesting parts → Nest
出现对话框, 然后点击 By name (零件名请根据建模者提供的零件表来进行操作)
- 5: 查看零件信息
路径: Nesting → Shop drawing information → Production information
出现一个对话框: (这里我们只说几个重要的选项)



- I. Bevel → 查看某一零件的坡口信息 (可以通过鼠标来指定需要查看的零件)
- II. Excess → 余量信息
- III. Side information → 方向信息
- IV. Part name → 零件名
- V. Panel name → 板架名



1: 加起割点

路径: Nesting → Nesting tools → Start → Burn

(制作切割数据的时候, 我们一般选用 single 根据相关规定对板内需要加切割点的地方进行相关操作, 前涉到的内容比较多, 请询问相关高手)

2: 加零件划线

路径: Nesting → Nesting tools → Start → Mark

(一般选用 All mark, 对于个别需要特殊加工的零件, 划线应手动 single 加进去)

3: 加过桥

路径: Nesting → Nesting tools → Start → Bridge

(牵涉的内容过多, 另讲)

4: 交换切割点

路径: Nesting → Nesting tools → Start → Move

5: 查看零件是否超出所使用钢板的范围

路径: Nesting → Shop dwg I/O → Overlap check

(如果有零件显示为红色, 请检查零件与钢板边缘的距离)

6: 改变起割点的顺序

路径: Nesting → Nesting tools → Start → Sequence

④:



1: 改变图框比例

路径: (没找着.....·#¥#%—*.....%¥#!·##¥%%.....)

2: 通过所给数值对图层进行移动

例: 向右移动 100 的位置

100 + 空格 + R (Right 英文缩写) 其他类型以此类推

3: 通过 2 点移动

4: 通过输入文字, 对所选图层进行旋转

5: 通过 2 点对所选图层进行旋转

- 6: 使所选图层, 按逆时针方向旋转 45 度
- 7: 使所选图层, 按逆时针方向旋转 90 度
- 8: 使所选图层, 按逆时针方向旋转 180 度
- 9: 使所选图层, 按顺时针方向旋转 90 度
- 10: 对所选图层, 左右镜象
- 11: 对所选图层, 上下镜象
- 12: 使所选图层按所给线段镜象
- 13: 平行于所参照的线段
- 14: 选择参照物
- 15: 垂直移动
- 16: 水平移动

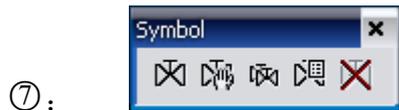


- 1: 确定 (O 键或回车键)
路径: Zap → Operation Complete
- 2: 结束 (Q 键)
路径: Zap → Quit
- 3: 取消 (R 键或 ESC)
路径: Zap → Cancel
- 4: 属性 (P 键)
路径: Zap → Options
- 5: 全部 (A 键)
路径: Zap → All
- 6: 刷新 (T 键)
路径: View → Repaint
- 7: 自动回到主画面 (S 键)
路径: View → Zoom → Auto
- 8: 放大 (Z 键)
路径: View → Zoom → In
- 9: 缩小 (D 键)
路径: View → Zoom → Out
- 10: 移动窗口 (W 键)
路径: View → Pan



这些快捷键是系统管理人员根据工作者要求来进行设置的, Tribon 原没有此功能, 各个公司也有不同

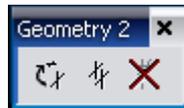
相当重要



⑦:

1 2 3 4 5

- 1: 选择符号
路径: Insert → Symbol
- 2: 移动符号
路径: Modify → Move → Symbol
- 3: 拷贝符号
路径: Modify → Copy → Symbol
- 4: 改变符号的属性
路径: Modify → Properties → Symbol
- 5: 删除符号
路径: Edit → Delete → Symbol



⑧:

1 2 3

- 1: 可以移动图面上的任何东西
路径: Edit → Transform → Geometry
- 2: 可以复制任何东西
路径: Modify → Copy → Geometry
- 3: 可以删除任何东西
路径: Edit → Delete → Geometry

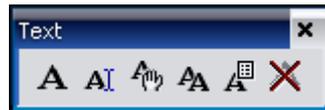


⑨:

1 2 3 4 5

- 1: 加入下划线
路径: Insert → Underline
- 2: 加入图片
路径: Insert → Subpicture
- 3: 加入零件
路径: Insert → Plate Part
- 4: 加入图框
路径: Insert → Drawing Form
(熔盛的图框名字为 NRS* 然后选择相对应的图框)
- 5: 输入多行文字
路径: Insert → Text,multi-line

⑩:



1 2 3 4 5 6

- 1: 加入文字 (英文字体格式一般为 TBISOFont101)
路径: Insert → Text,single-line
- 2: 修改文字
路径: Edit → Text
- 3: 移动文字
路径: Modify → Move → Text
- 4: 拷贝文字
路径: Modify → Copy → Text
- 5: 修改文字信息
路径: Modify → Properties → Text
- 6: 删除文字
路径: Edit → Delete → Text

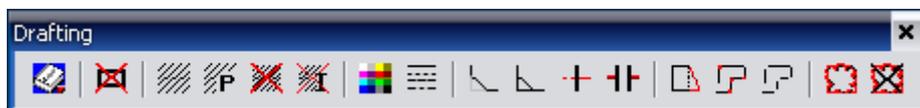
⑪:



1 2 3 4 5 6

- 1: 选择图层
路径: Tools → Subpicture → Current
- 2: 移动图片
路径: Modify → Transform → Subpicture
- 3: 拷贝图片
- 4: 删除图片
路径: Edit → Delete → Subpicture
- 5: 分开图片
路径: Tools → Subpicture → Split
- 6: 合并图片
路径: Tools → Subpicture → Regroup

⑫:



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

- 1: Log Viewer (牵涉内容较多, 另行介绍)
路径: View → Log Viewer
- 2: 删除某个指定区域内的内容 (这个区域由操作者自己决定)
路径: Edit → Delete → By Area
- 3: 增加阴影
路径: Insert → Hatch Pattern

4: 选择阴影格式

路径: Format → Hatch Pattern

5: 删除阴影

路径: Edit → Delete → Hatch Pattern

6: 使阴影内的文字不被阴影所覆盖

路径: Edit → Delete → Hatch Pattern Island

7: 改变颜色

路径: Modify → Colour

8: 改变线型

路径: Modify → Line Type

9: 倒角 1

路径: Modify → Trim → Fillet

倒角的样式有以下三种, 根据需要选择:



例如:



10: 倒角 2

路径: Insert → Fillet

倒角的样式有以下三种, 根据需要选择:



例如:



11: 截断 (或延伸)

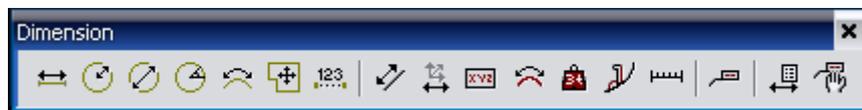
路径: Modify → Trim → To Entity (使用方法问高手)

12: 一分为二

13: 移动一个范围内的东西

- 14: 使多个有连接的线段连接成为一条
- 15: 断开多段线
- 16: 云线（醒目标注）
- 17: 删除云线

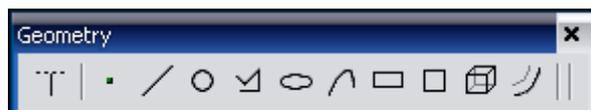
⑬:



- 1
- 2
- 3
- 4
- 5
- 6

- 1: 标注
- 2: 测量圆的半径
- 3: 测量圆的直径
- 4: 测量角度
- 5: 测量曲线的长度
- 6: 测量两点之间的距离

⑭:



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

- 1: 点
- 2: 线段
- 3: 圆
- 4: 多段线
- 5: 椭圆
- 6: 曲线
- 7: 多种矩形
- 8: 正方形
- 9: 功能另外介绍
- 10: 平行线

⑮:



- 1
- 2
- 3
- 4
- 5
- 6
- 7

- 1: 新建
- 2: 打开
- 3: 保存
- 4: 剪切
- 5: 复制
- 6: 粘贴

附录：

这里转发一篇 Tribon 公司相关人员所提供的系统用户界面开发的文章，希望对大家有点帮助，看不懂的可以打电话问我，能帮肯定帮！有兴趣的朋友可以自己研究一下，具体如下：

Tribon MX 系统用户界面开发

摘要：本文主要讲述了 Tribon MX 系统用户界面开发的原理、过程及其在船体设计中的应用。

关键词：计算机应用 Tribon 系统 Vitesse 二次开发

1 前言

Tribon MX（包括 M1、M2、M3）系统操作界面及其命令的输入方式主要有三种：Menu（菜单栏）、Tool bar（工具条）及 Accelerator（快捷键），三种方式各有优缺点。就船体应用的各模块（有 Drafting、Planar Hull 和 Curved Hull）而言，工具条和快捷键的操作相对较为便捷。但 Tribon 系统本身提供的这两类快捷方式不够完善：有些常用的命令并没有提供快捷方式，如 Dimension_Distance to plane leader；而有些命令虽然设置了快捷键，但是实际操作起来并不“快捷”，如 Zoom in 命令的快捷键 Ctrl+I，笔者用的时候要左右手同时用上，很不方便。针对这种情况，笔者借助 Tribon MX 系统的 Vitesse 程序接口开发了一些船体建模和绘图常用命令的工具条和快捷键；并对系统原有的工具条和快捷键进行了优化，删除了一些使用率偏低的命令的快捷方式。

2 开发原理及过程

Tribon MX 各基于 Drafting 的应用模块在启动时，系统通过 Vitesse 程序接口读取存放在环境变量 SBB_TRIGDIR 所指向目录的初始化文件 trig_draft_init.py 以加载用户自定义操作界面。因此开发的主要工作就是按照用户需要编写初始化文件，并定义环境变量 SBB_TRIGDIR。

2.1 Vitesse 程序简介

Vitesse 程序是 Tribon MX 系统提供的用户二次开发工具。其通过读取用户编写的程序文件，可以加载用户自定义操作界面，包括菜单栏、工具条和快捷键，也可以新建和修改图形，甚至可以新建和修改船体及舾装模型，而这一切都是自动的，不需要用户过多的参与。例如通过运行一个 Vitesse 程序，可以实现系统自动生成板架，自动创建扶强材并自动处理端部形式，自动定义开孔的位置及大小，自动增加肘板等，而不需要先新建板架，再创建构件，最后再保存模型等一系列工作。

Vitesse 程序使用的编程语言是解释性面向对象语言——Python。

2.2 初始化文件

trig_draft_init.py 文件是 Tribon 系统内置的一个 Drafting Trigger，它不能通过 Vitesse 程序中的 Run Script 命令运行，而只能在任何一个基于 Drafting 的模块（如：Planar Hull、Basic Design、Pipe Modelling 等）启动时运行。

Tribon 系统要求用户自定义界面的所有命令都须定义在 Drafting post-trigger 函数中，而整个用户界面初始化文件实际上就是一个 Drafting post-trigger 函数的定义文件。此文件的格式是 Python 程序源文件格式，可以用记事本或 Python 编辑器编辑。其语法结构通常如下：

```
import <module name1>
```

```

import <module name2>
:
import <module namen>
def post():
    try:
        :
    except:
        :
    return kcs_util.trigger_ok()
post()

```

这里需要注意的是定义的函数名一定要为 post，否则此 trigger 将不起作用。

1) module (模块)

在 Python 语言中 module 通常可以看作是多个函数的集合。Tribon 系统中提供了许多这样的模块，而本次开发作用到的主要有 kcs_gui、kcs_util 和 basic_design_menu。另外为了方便今后程序的改进与更新，笔者把工具条和快捷键的命令分别存放在 toolbar 和 accelerator 模块中。

2) 系统模块

kcs_gui 模块包含了所有 Tribon 系统内置的用户界面命令，如菜单栏的增减，工具条的修改，快捷键的设置等命令。因此，要想使用用户界面命令自定义界面就必须 import 此模块。

kcs_util 模块则包含了返回 trigger 码，访问 Tribon 系统环境变量，检查操作是否正常运行等命令。语句 return kcs_util.trigger_ok() 的作用就是当用户界面初始化正常结束时告知用户已加载成功。

basic_design_menu 模块的作用是在 Basic Design 程序中加载 Quick Panel 菜单。

3) 用户模块

用户自定义模块的文件格式同样也是 Python 程序源文件格式，其语法结构同 trig_draft_init.py 相似，只不过函数的命名不同，注意不要用 post、pre 等 Tribon 系统内置的函数名即可。如 toolbar 模块中定义了 modify、planar、curved 等函数，accelerator 模块定义了 add 函数；用户界面的命令就分别设置在这些函数当中。

用户模块文件应存放在 Tribon 系统环境变量 SBB_TRIGDIR 或者操作系统环境变量 PYTHONPATH 所指定的目录当中。

2.3. 本次开发所涉及到的用户界面函数及参数如下：

1) 命令及工具栏 ID

Tribon 系统内每个 function (命令) 和 tool bar (工具栏) 都对应有 ID 号，这个号在同一个 Tribon 系统版本中是唯一的、固定的。

当 Tribon 的应用程序第一次运行时，系统会自动生成一个文件名为 kcs_gui_<application>.py 的 Vitesse script 文件，并将其保存在环境变量 SB_PYTHON 所指定目录内的 lib 子目录中。而这个文件正是命令和工具栏的 ID 定义文件。

2) tool bar 函数

·toolbar_get(Id)

——提取系统原有工具条，Id 为工具条在 ID 定义文件中所赋予的数值。

·toolbar_add(Caption)

——新增工具条，Caption 为工具条的名称，注意要加双引号，如"View"。

·toolbar_button_std_add(Toolbar, Position, Function, ImageFile, Tooltip, Message)

——在工具条中新增快捷按钮，Toolbar 为工具条名称，Position 为新增按钮在工具条中的位置号，Function 为按钮所代表命令的 ID 号，Imagefile 为按钮图标文件的绝对路径，Tooltip

为鼠标悬浮于按钮时的提示信息, Message 为鼠标悬浮于按钮时系统状态栏所显示的信息。

·toolbar_button_remove(Toolbar, Position)

——移除工具条中的快捷按钮, Toolbar 为工具条名称, Position 为移除按钮在工具条中的位置号。

3) accelerator 函数

·accelerator_add(Key, SystemKeys, Function)

——Key 为用户自定义按键的名称, 只允许有小键盘区域的各键和 F1—F12 键; SystemKeys 为组合键, 0 代表无组合键, 1 代表 ALT 键, 2 代表 CTRL 键, 3 代表 SHIFT 键; Function 为命令的 ID 号。

·accelerators_obsolete_add()

——此函数的作用是使系统调用 TB5 方式快捷键。

2.4. 环境变量设置

为了便于管理与更新, 笔者将初始化定义文件和用户自定义模块文件都存放在 Tribon 系统服务器 tbm1s1 标准文件目录 TBStdDb 的子目录 Trigger 中。因此只要将 Tribon 系统环境变量 SB_TRIGDIR 赋值 P:\TBStdDb\Trigger 即可。

3 界面加载

对于设计人员而言, 用户界面的加载不需要过多的干预, 在选择工程项目之后, 启动船体应用各模块时, 系统会自动加载自定义的快捷键和修改过的快捷工具条, 新增加的工具条则需在 View 菜单下的 User Toolbars 里选择。

这里需要注意的是, 由于目前笔者只针对船体用户进行了开发, 用户只有在选择各工程项的 HULL 子项启动船体应用模块时方能有效, 对于其他子项则保持系统默认原样不变。

自定义的工具条和快捷键见附图。

4 结束语

到目前为止, 此项开发已全面应用于 M2 系统的 29000t、38500t、51800t 和 52500t 船以及 M3 系统多条船的船体生产设计建模、绘图中。从实际使用情况来看, 用户反映操作较为方便, 效果较为明显, 普遍提高了建模绘图的速度和设计的效率。

附图

1. 自定义工具条：删除了一些基本不用的快捷图标，并新增以下命令：



1

1: Print preview



1 2 3

1: General restriction symbol 2: Cloud 3: Insert hatch pattern



1 2 3 4 5 6

1: Chain 2: Dechain 3: Trim to entity 4: Trim by length 5: Trim gap 6: Trim fillet



1

1: Underline



1 2 3 4

1: Distance to plane leader 2: Note pos no 3: Delete dimension 4: Delete position number



1 2 3 4

1: View Create 2: View Detail 3: View Recreate 4: View Modify



1

1: Scheme edit



1 2 3 4 5 6 7

1: Create Curved Panne 2: Create hull curvel 3: Create Feature/Hole 4: Create Feature/Notch 5: Curved/view/developed plate 6: Hull tools/dimensioning 7: Hull

tools/ppi hull

2. 小键盘区域快捷键:

命 令		小键盘区对应键	TB5 方式主键盘区对应键
Zap	Operation Complete	Enter	O
	Quit	• (Del)	Q
	Cancel	0	R
	Options	+	P
	All	-	A
	Subpicture Lever1	1	1
	Subpicture Lever2	2	2
Subpicture Lever3	3	3	
View	Repaint	7	T
	Zoom In	6	Z
	Zoom Out	4	D
	Zoom Auto	5	S
	Zoom Previous	9	Ctrl+E
	Pan	8	W

嘿嘿……他结束了我没结束，不好意思！我再加点需要文件的具体路径,希望各位能自己研究懂:

1: D:\toolbar\icon

TRIBON 快捷键图标所保存的绝对路径

2: D:\Tribon\M3\Vitesse\Lib\kcs_gui_*****.py

TRIBON 所有操作命令的函数的定义文件，所有的命令代码都在这里边

3: D:\Tribon\M3\Vitesse\Trigger\ trig_draft_init.py

可以通过这个文件来控制用户自定义快捷键文件的路径

4: 用户自定义文件存放格式应为:

建模 : ****_PLANAR_HULL.py

套料 : ****_NEST.py (具体编辑格式看附录 2)

以此类推…… “****” 可以用自己名字，这样才个性，呵呵……

此文件名称定义好后，请在 trig_draft_init.py 里将相对应的文件名称改为一致!!!

具体操作，上面那位高手已经介绍的很清楚了，看你们自己的了!! 就这么多!!

再见!!!

张 宾

2007 年 1 月 24 日

附录2

```

import kcs_gui
import kcs_util

try:
    my_toolbar = kcs_gui.toolbar_add("zb-nesting-1")
    my_toolbar2 = kcs_gui.toolbar_add("zb-nesting-2")
except:
    print "Failed creating tool bar", kcs_gui.error

#
# Add the following buttons to the toolbar:
#
# Open Volume...
# New Volume...
# -----
# My command 1
# My command 2
#

try:
    kcs_gui.toolbar_button_std_add(my_toolbar, 1, 32888, "D:\\toolbar\\icon\\outfit\\Note_Delete.ico")
    kcs_gui.toolbar_button_std_add(my_toolbar, 2, 32977, "D:\\toolbar\\icon\\outfit\\Area_Delete.ico")
    kcs_gui.toolbar_button_std_add(my_toolbar, 3, 32915, "D:\\toolbar\\icon\\outfit\\NoFillet.ico")
    kcs_gui.toolbar_button_std_add(my_toolbar, 4, 32942, "D:\\toolbar\\icon\\outfit\\Insert_Subpic.ico")
    kcs_gui.toolbar_button_std_add(my_toolbar, 5, 33764, "D:\\toolbar\\icon\\outfit\\Slice.ico")

    kcs_gui.accelerator_add("F2", 0, 32772) # Insert Line #
    kcs_gui.accelerator_add("F3", 0, 32773) # Insert Arc #
    kcs_gui.accelerator_add("F4", 0, 32857) # insert_symbol #
    kcs_gui.accelerator_add("F5", 0, 32864) # insert_text_singleline #
    kcs_gui.accelerator_add("F6", 0, 33087) # planar_scheme_edit #
    kcs_gui.accelerator_add("F7", 0, 33432) # hull_tools_ppi_hull #
    kcs_gui.accelerator_add("F8", 0, 32874) # dimension_linear #
    kcs_gui.accelerator_add("F9", 0, 32875) # dimension_radius #

    kcs_gui.accelerator_add("NUMPAD0", 0, 33775) # tools_model_view_autoscale_all #
    kcs_gui.accelerator_add("NUMPAD1", 0, 34409) # viewport_new_shaded_viewport #
except:
    print "Failed adding toolbar buttons", kcs_gui.error

```

Tribon 内显示的工具条名称

命令代码

快捷键图标的路径

F2-F12快捷键设置

小键盘快捷键设置