Tribon Hull Manager

Training Guide • Training Guide • Training Guide

Revision Log

Date	Page(s)	Revision	Description of Revision	Release
02/062004	All	SK	General update for M3	M3

Updates

Updates to this manual will be issued as replacement pages and a new Update History Sheet complete with instructions on which pages to remove and destroy, and where to insert the new sheets. Please ensure that you have received all the updates shown on the History Sheet.

All updates are highlighted by a revision code marker, which appears to the left of new material.

Suggestion/Problems

If you have a suggestion about this manual, the system to which it refers, or are unfortunate enough to encounter a problem, please report it to the training department at

Fax +44 191 201 0001
Email training@tribon.com

Printed by Tribon Solutions (UK) Ltd on 17 January 2005

Contents

Chapter 1

1 Initialising a Tribon Project

After copying the relevant directories and files and editing the d065 file it is now necessary to initialise the Form data bank (**SB_CGDB**) and the Structure data bank (**SB_OGDB**). This involves the creation of various objects and tables in both the **SB_CGDB** and **SB_OGDB**. These objects and tables will inform the Tribon system which hull form to use, what the frame spacing is, what the prefix for the naming of seams and butts should be, etc, etc…

This object and table creation is done using the Tribon M3 Initiate Hull Standards utility.

Use the **Project Selection** tool and select your new project.

Start the utility by clicking on **Start Menu > Programs > Tribon M3 > Hull > Initiate Hull Standards**

In the tree structure on the left-hand side of the resulting application expand the **Initiate Hull Model** node and the following nodes will be displayed.

Hullref, create	Creates a Hull Reference Object in the SB_CGDB
Structref, create	Creates a Structure Reference Object in the SB_OGDB
Blocks, manipulate	Creates Block objects in the SB_OGDB
Frame/long positions, create	Creates an __SBH_GENTAB__ object in the SB_OGDB

The purpose and use of the above four functions are explained in the following chapters.

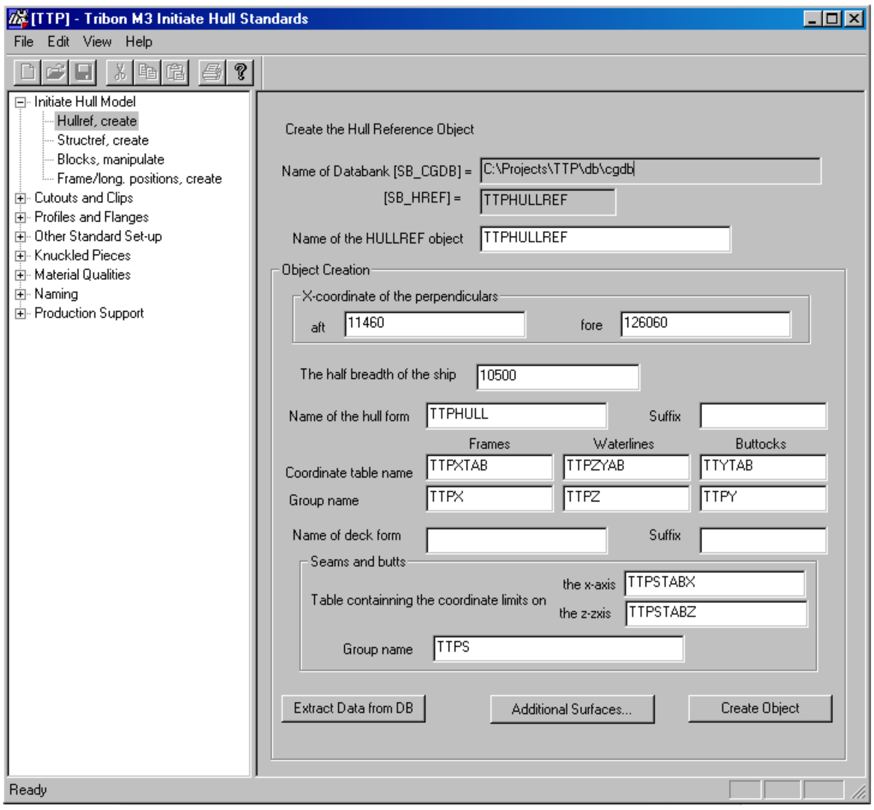
1.1 Initialising the Form data bank (SB_CGDB)

The form databank contains information related to the surfaces of the ship i.e. the ship surface itself and curve information derived from the surfaces. Additionally, there are some tables that keep a record of the names of objects in this data bank and of the objects stored there.

1.1.1 Creating a Hull Reference Object

The hull reference object is a small table containing information about names and name rules of objects in the form databank (**SB_CGDB**). The Tribon modules access the name of this object via the Tribon environment variable **SB_HREF**.

The hull reference object is created or modified by the **Hullref, create** option. Clicking on this option will result in the following form being displayed:



Name of Databank (SB_CGDB) and **(SB_HREF)** will be automatically filled in by the system assuming the relevant variable is set in the current project.

Name of HULLREF object: An arbitrary string, but usually a combination of the project identification, (ship letters) and the word HULLREF. This field will also be automatically filled in with the current value of SB_HREF. This allows either the modification of this existing object or the creation of a new hull reference object. Multiple hull reference objects can exist in one project but only the one currently assigned to SB_HREF will be read by the system.

X co-ordinate of the perpendiculars: The relevant X co-ordinate for the Aft and Fore Perpendiculars (given in mm)

The Half breadth of the ship: The half breadth of the ship (given in mm)

Name of the hull form: If using TID software to produce the hull form this name should match the name of the DML or DM file stored in the directory in which the Surface Server was started.

Suffix: Curves created in these additional surfaces are named according to the same rules as curves in the main surfaces. To separate them from the main surface curves the group names of these additional surfaces have an additional "suffix" by which the group name will be extended.

Example of Composed Names of Objects in Multiple Surfaces:
In order to allow the same numbers to be used for objects in different surfaces, it is necessary to specify a surface specific extension of the group names. This surface suffix consists normally of one letter.
e.g. Suppose that there is a seam with number 123 in an additional surface with surface suffix C and that the group name for seams is AAS. Then the name of that seam will be AASC123.

The main hull need not have any surface suffix.

Co-ordinate table name: The name of the co-ordinate tables for frame, waterline and buttock curves. These tables contain the X, Y and Z co-ordinate of the plane in which the curve with a given number is located.

Group name: The group names of frame, waterlines and buttocks. The names of these main curves are composed by a "group name" concatenated with a curve number (e.g. a frame number). These group names are defined in this object.

Name of deck form: If using TID software to produce the deck form this name should match the name of the DML or DM file stored in the directory in which the Surface Server was started. If no deck form is present in this particular project, the field should be left blank.

Seams and butts:

Table of co-ordinate limits on X-axis/ Z-axis: The names of the limit tables along the X and Z axes for seams. One of these tables contains the minimum and maximum co-ordinates along the X axis of all seams, the other the same information for the Z axis.

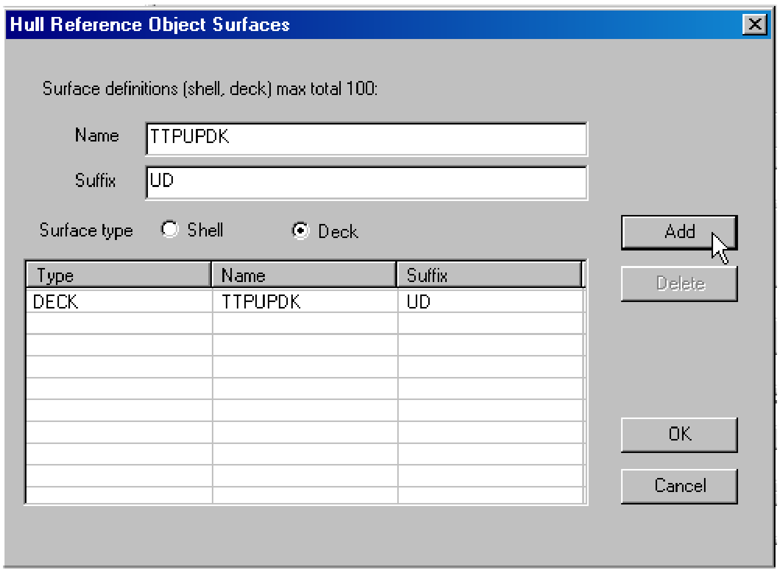
Group name: Defines prefix to be given to Seams and Butts in the tables above

Additional Surfaces: Opens the dialog box shown opposite, where the user can add up to 100 additional surfaces.

Name: If using TID software to produce the additional surface this name should match the name of the DML or DM file stored in the directory in which the Surface Server was started.

Suffix: Curves created in these additional surfaces are named according to the same rules as curves in the main surfaces. To separate them from the main surface curves the group names of these additional surfaces have an additional "suffix" by which the group name will be extended

Surface type: Select either **Shell** or **Deck**



Add: After completing the 3 fields above use this button to submit the additional surface information.

Delete: Highlight an existing additional surface in the list displayed and use this button to delete it.

OK: Use this button to exit the function after Adding/Deleting the desired surfaces.

Extract data from DB: The system will refresh the current form with the latest data from the SB_CGDB.

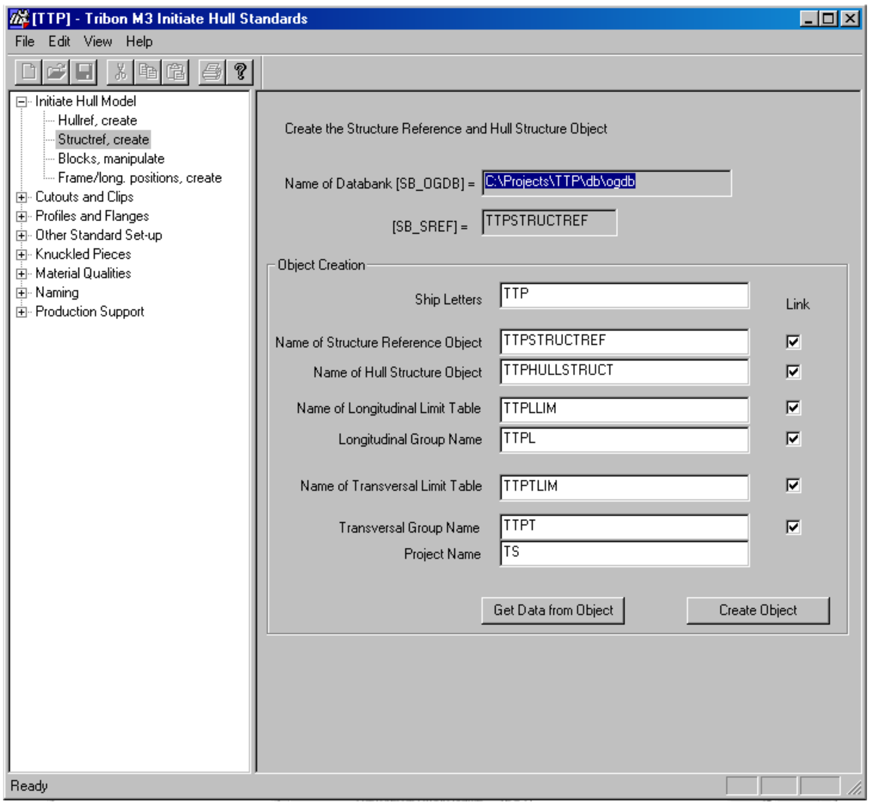
Create Object: After completing all the required fields use this button to create/update the hull reference object in the SB_CGDB.

1.2 Initialising the Structure data bank (SB_OGDB)

The structure databank contains model information about the internal structure of the ship. The model information in the structure information is stored according to nominal dimensions and the adjustments for production are made when parts are extracted for production. Examples of such adjustments are shrinkage compensation, excess, shell plate development, development of knuckled pieces, changes for (varying) bevels angles and bevel gaps.
Additionally the structure data bank contains some table information and objects that describe miscellaneous types of hull standards, set up by the customer.

1.2.1 Creating a Structure Reference & Hull Structure Object

The structure reference object is a small table containing information about names and name rules of objects in the structure databank. The Tribon application modules access the name of this object via the Tribon environment variable **SB_SREF**.
The structure reference object is created or modified by the **Structref, create** option. Clicking on this option will result in the following form being displayed:



Name of Databank (SB_OGDB) and **(SB_SREF)** will be automatically filled in by the system assuming the relevant variable is set in the current project.

Ship Letters: It is suggested that the names of all hull objects of a certain project should start with the same one or two letters. Check all of the **Link** boxes and the letters keyed into this field will be automatically added to the default names for the other objects in this menu.

Name of Structure Reference Object: Usually ship letters + STRUCTREF

Name of Hull Structure Object: Usually ship letters + HULLSTRUCT. The Hull Structure object is the object that serves as the entry to the hull model via the design structure. It does not contain any relevant information except the references to all the blocks. The Hull Structure object is automatically updated each time a block object is created, modified or deleted. The designer never really gets in direct contact with the Hull Structure object.

Name of Longitudinal Limit Table: The name of the extension table for longitudinals along the X axis (min-max co-ordinate values). The names of limit tables for the extension along the Y and Z axes are formed by adding 'Y' and 'Z', respectively, to this name.

Longitudinal Group Name: The group names of longitudinals. The name for one of these objects is created by a "group name" plus a number added in Curved Hull or Basic Design, typically the longitudinal position multiplied by 10.

Name of Transversal Limit Table: The name of the extension table for transversal frames along the X axis (min-max co-ordinate values). The names of limit tables for the extension along the Y and Z axes are formed by adding 'Y' and 'Z', respectively, to this name.

Transversal Group Name: The group names of transversal frames. The name for one of these objects is created by a "group name" plus a number, typically the relevant frame number.

Project Name: The project name of the current project. This name may be used as a part the production oriented part names (and may thus be considered as an "external" correspondence to the ship letters that are for internal use).

Get data from Object: The system will refresh the current form with the latest data from the SB_OGDB.

Create Object: After completing all the required fields use this button to create/update the structure reference object in the SB_OGDB.

1.3 Defining Frame / Longitudinal positions

Within Tribon it is possible to define an object that contains the entire frame and longitudinal position information for the current project. Within this object it is possible to define both horizontal and longitudinal grid positions i.e. distances from the centreline and also vertical longitudinal positions i.e. distances from the baseline.

The object will be named **__SBH_GENTAB__** and will be stored in the structural database (**SB_OGDB**). The object is very important within a Tribon project as many of the applications use this object to calculate the position of model objects that are located using frame or longitudinal position references.

Before discussing the creation of the object a few Tribon numbering rules should be considered.

1.3.1 Frame numbering in Tribon

1. The frames must be integers i.e. they must not contain any letters, however they may be negative.
2. The number of the frames should be in the range [-899,2276]
3. The maximum number of frames is currently restricted to 500, unless the frames are consecutively numbered. In the latter case the frames may have numbers in the range [-99,500], i.e. 600 in total.
4. The relation between frame number and frame position may be quite arbitrary, e.g. they may be increasing with increasing x-co-ordinates, decreasing with increasing x-co-ordinate or set without any specific order with relation to the frame position.
5. The distance between frames may vary arbitrarily.

It is common within shipbuilding to locate frame number 0 at the aft perpendicular and to let the frames in the aft peak be identified by letters; A, B, C, etc. The rules above do not allow this denomination.

It is recommended that the letters be replaced by negative numbers (A→ -1, B→ -2, etc.).

In some regions of the world it is customary to have numbered frames only at web frames and to identify intermediate frames by adding letters to the main frame number, e.g. 56, 56A, 56B ...., 57, 57A, 57B, ... It is recommended that the letters in the example are replaced as follows; 56, 561, 562,...., 57, 571, 572, ... (or to 560, 561, 562, ..., 570, 571, 572, ...).

1.3.2 Longitudinal position numbering in Tribon

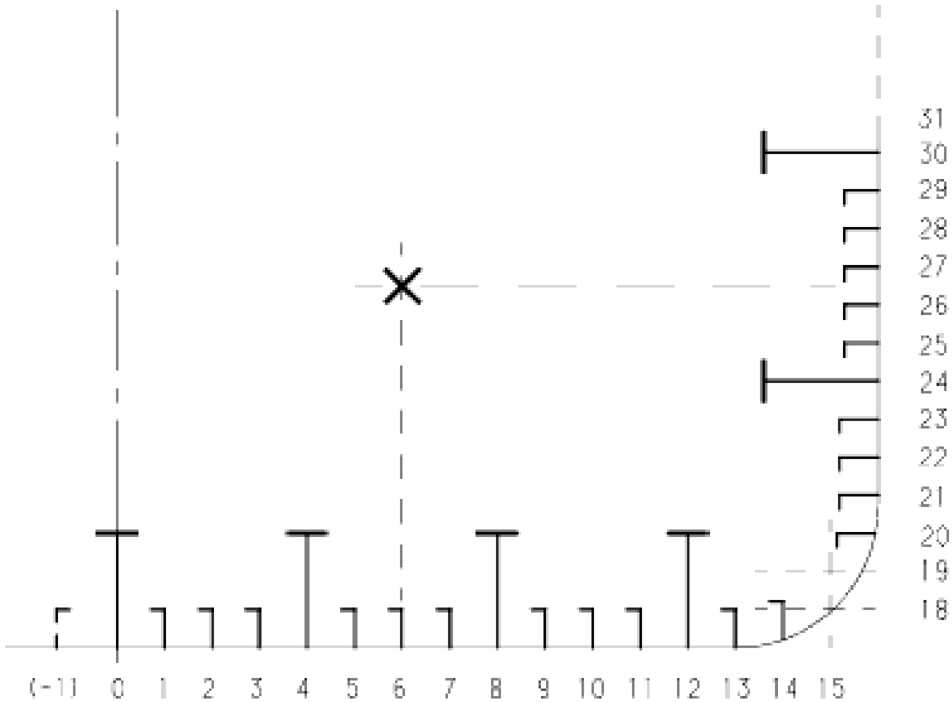
Frame positions are in most cases defined at those locations along the ship where there are transversal hull members, either frames or webs, etc. In a similar way there are in most ships characteristic distances from the Centre Line (CL) and above the Base Line (BL) where hull members are located. E.g. longitudinals in the bottom and in the side in the midship section are located at positions which normally also define the position of stiffeners in decks, platforms, bulkheads, etc, and the position of girders. By referring to these positions one may define locations along the Y and Z axes as simple as e.g. Y=LP10 +100 and Z=LP35 -100. (LP10 +100 means 100mm in portside direction from Longitudinal Position number 10 in the bottom, LP35 -100 means 100 mm below Longitudinal Position 35 in the side).
From a practical point of view it is recommended to let the longitudinal positions and their numbers coincide with the numbers and positions of actual longitudinals in the midship section. However, it should be noted that the longitudinal positions form a grid that need not have any direct relation with the physical longitudinal frames. E.g. if some longitudinals are replaced by girders there are "holes" in the numbering of longitudinals. However, the longitudinal positions should include all the positions, also those where there are no longitudinal frames.

The figure below shows schematically a typical midship frame with suggested longitudinal positions.

The point at the cross in the figure above may be located by Y=LP6, Z=LP26.5

The following rules should be considered:

1. The positions and the numbers should be related to those of actual longitudinal frames, if possible.
2. The longitudinal numbers should be in the interval [0,999]
3. The numbers for horizontal positions (along the Y axis) and vertical positions (along the Z axis) should not be the same.



4. It is quite possible to define a longitudinal position in the CL plane, i.e. where Y=0. This position may have number 0.

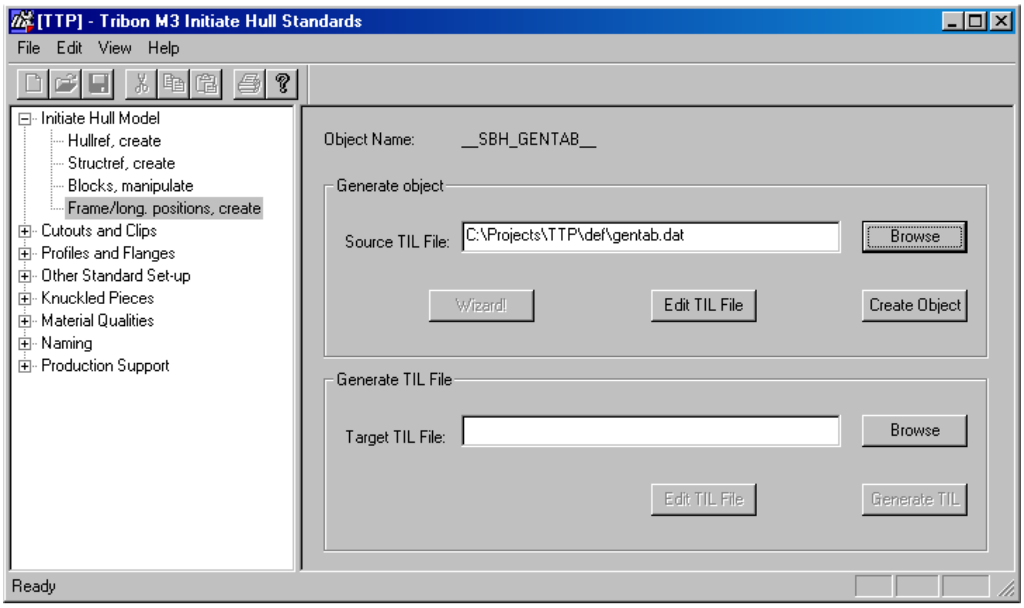
5. The relation between increasing/decreasing numbers and increasing/decreasing distances is arbitrary similar to what is stated for frames. This should be decided by the rules for longitudinal numbering, used by the yard.

6. There is no direct connection between the longitudinal position numbers and the generated physical longitudinal frames.

7. Longitudinal positions in the bottom are normally only defined on portside. Reference to the corresponding positions on the starboard side is done by negating the longitudinal number, e.g. Y=LP-20+100.

1.3.3 Creating the SBH GENTAB object

The **__SBH_GENTAB__** object is created or modified by the **Frame/long positions, create** option. Clicking on this option will result in the following form being displayed:



The **__SBH_GENTAB__** object is created by the system reading a suitable TIL file.

If an input file already exists then use the **Browse** button in the **Generate object** field to locate the file.

If no file exists then use the **Generate TIL** button in the **Generate TIL File** field to create a new file. The system will prompt for a name for the file and it should then be saved before exiting the editor. After creating the file use the **Browse** button in the **Generate object** field to locate the file.

When the file has been located successfully in the **Generate object** field, use the **Edit TIL File** button to open the file with the default Windows editor and this allows editing to suit.

After the successful editing of the file close and save it. Click the **Create object** button and the system will generate the __SBH_GENTAB__ object.

1.3.4 The contents of the TIL file

The input file is organised in "record types" with layout as described below. The format is free but it is recommended to have one record per line. The line width is limited to 80 characters. The number of records is unrestricted.

Record Type 0

This record must specify the name of the current structure reference object.

E.g. **0 ''<NAME>STRUCTREF**

The line should consist of the digit zero followed by a blank space, then a single apostrophe followed immediately by another single apostrophe then a blank space followed by the name of the structref object.

Record Type 2

This record has no parameters. If it is included in the input file the system will produce an output file containing all of the frame and longitudinal positions generated along with their corresponding co-ordinate value. It is recommended that this record type is always included.

Record Type 20

This record type informs the system of the desired frame number and position.

E.g. **20 START STEP END COORD COORDSTEP**

START	The first frame number for which to add or change a co-ordinate
STEP	The difference in frame numbers for the current record
END	The last frame number for which to add or change a co-ordinate
COORD	The co-ordinate for the frame **START**
COORDSTEP	The distance between each frame in the range **START** → **END**

Recommended layout for record type 20 is as follows:

The first line must contain information for a single frame only (usually FR0)
The next line(s) should contain information for any negative frame numbers
The next line(s) should contain information for any positive frame numbers
The next line(s) should contain information for any ice frames (positive and negative)

Record Type 30

This record type informs the system of the horizontal longitudinal positions in the ship's bottom and their positions relative to the centreline. The records must be given such that the co-ordinates are in strictly ascending or descending order.

E.g. **30 START STEP END COORD COORDSTEP**

START	The first longitudinal for which to add or change a co-ordinate
STEP	The difference in longitudinal numbers for the current record
END	The last longl number for which to add or change a co-ordinate
COORD	The co-ordinate for the longitudinal **START**
COORDSTEP	The distance between each longitudinal in the range **START** → **END**

The longitudinal numbers should not be multiplied by 10 and they have to be equal to or greater than 0.

Record Type 40

This record type informs the system of the vertical longitudinal positions in the ship's side and their positions relative to the base line. The records must be given so that the co-ordinates are in strictly ascending or descending order.

E.g. **40 START STEP END COORD COORDSTEP**

START	The first longl number for which to add or change a co-ordinate
STEP	The difference in longitudinal numbers for the current record
END	The last longl number for which to add or change a co-ordinate
COORD	The co-ordinate for the longitudinal **START**
COORDSTEP	The distance between each longitudinal in the range **START** → **END**

The longitudinal numbers should not be multiplied by 10 and the first number should be greater than the final horizontal position number defined in record type 30.

Example of input

gentab.dat - Notepad

File Edit Format Help

```
0 '' TTPSTRUCTREF
2
20 1 1 20 8310 850
20 21 1 40 25160 700
20 41 1 100 38885 425
20 101 1 122 64500 500
20 123 1 214 75600 600
30 0 1 4 0 500
30 5 1 10 2510 640
30 11 1 16 6390 680
40 19 1 37 1400 700
```

Resulting frame positions:

FR1is at X=8310, frame numbers then increase in steps of 1 until FR20 is reached with each frame being 850mm from the previous one
FR21 is then at X=25160, the frame numbers then increase in steps of 1 until FR40 is reached with each frame being 700mm from the previous one.
FR41 is then at X=38885 the frame numbers then increase in steps of 1 until FR100 is reached with each frame being 425mm from the previous one.
FR101 is at X=64500 and the frame numbers increase in steps of 1 until FR122 is reached with each frame being 500mm from the previous one.
FR123 is at X=75600 and the frame numbers increase in steps of 1 until FR214 is reached with each frame being 600mm from the previous one.

Resulting horizontal longitudinal positions:

LP0 is at Y=0 and the longitudinal position numbers then increase in steps of 1 until LP4 is reached with each longitudinal position being 500 from the previous one.
LP5 is at Y=2510 and the longitudinal position numbers then increase in steps of 1 until LP10 is reached with each longitudinal position being 640 from the previous one.
LP11 is at Y=6390 and the longitudinal position numbers then increase in steps of 1 until LP16 is reached with each longitudinal position being 680 from the previous one.

Resulting vertical longitudinal positions:

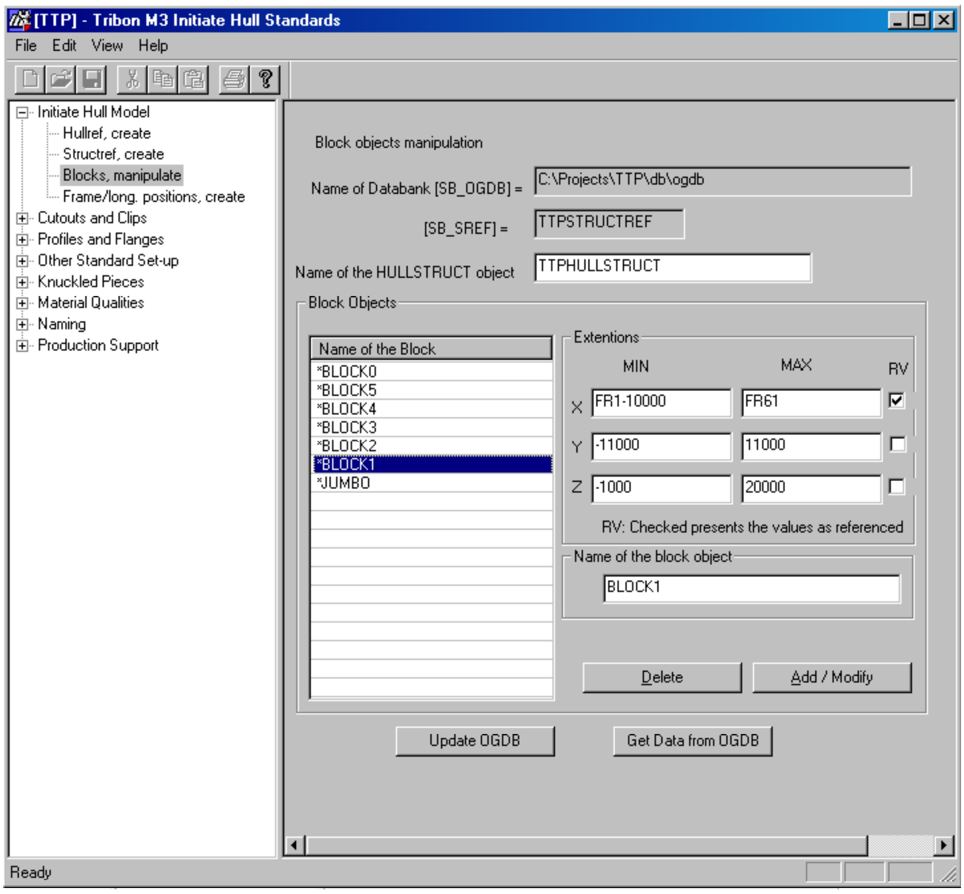
LP19 is at Z=1400 and the longitudinal position numbers then increase in steps of 1 until LP37 is reached with each longitudinal position being 700 from the previous one.

Additional increments may be added as required.

1.4 Creating a Block Object

Like the hull structure object the block objects do not carry any actual model information, only the location of its surrounding box in space. It should primarily be considered a geographically constrained container of panels, referred to from the Hull Structure object and itself referring to the panels that belong to the block. A block object is automatically updated each time a panel is created, modified or deleted.
The designer never really interacts with the block objects except when they are created. However, the block may be used as the "handle" by which information from the hull model is extracted in various situations.
The same block may include panels both on portside and on starboard. If a block is restricted to a side section (e.g. a side tank) its limits should be restricted to its limit on portside. Panels valid for the starboard side (and even those *modelled and stored* on starboard) may nevertheless belong to this block. Thus a block can always contain panels within its explicitly defined block but also panels within the box when mirrored in the centreline plane.
A block over the Centre line should be defined with its true limits.

A Block is created or modified by the **Blocks, manipulate** option. Clicking on this option will result in the following form being displayed:



This form serves as a view of the existing block objects in the databank and also enables the creation of new block objects.

Name of Databank SB_OGDB), (SB_SREF) and **Name of the HULLSTRUCT object** will be automatically filled in by the system assuming the relevant variable is set in the current project.

Name of the Block: A list of all existing blocks in the current project. This list also provides additional information as to whether this object can be found in the databank (marked with an asterisk – '*' in front of the name), or whether it is only referred to in the HULLSTRUCT object (denoted by an exclamation mark '!'). Clicking on a block name in the list will result in its particulars becoming current. Any of these values can then be edited and the block object updated.

Extensions: When a block is created it is necessary to define its (rough) extension ("box") in space. The panels belonging to a block should preferably be located entirely within this box. The boxes of different blocks may overlap. The representation of the values may be switched from FR/LP to absolute values by clicking the **RV** (referenced value) checkbox. If an existing block is selected, the values assigned will automatically populate the form.

Name of the block object: The name of the new block object to be created, or the existing block object to be modified. Automatically filled in if a block is selected from the list.

Delete Delete the current block object. Please note, if the block to be deleted contains panels these will also be deleted by this action.

Add/Modify: If an existing block has been selected from the list and it's extensions modified, or a completely new block name and extensions have been defined, this button will submit the new data.

Get Data from OGDB: The system will refresh the current form with the latest data from the SB_OGDB.

Update OGDB: Store any additions / modifications carried out to the blocks.

ⓘ *A block definition in Tribon does not have to reflect the building blocks used for construction. The use of the Assembly Planning Tool allows the actual build sequence to be completely redefined, regardless of the block definition in Tribon.*

2 Hull program defaults

After the installation of the software and the selection of a project, any of the hull programs can be started. It should be noted that default files control the appearance/behaviour of the majority of the applications. These files reside in the directory associated with the project variable **SB_SHIP**.

Unless you are doing Tribon work for various clients it is envisaged these default files will be set up once to suit your standards/requirements and referenced by all projects. If you are doing work for various clients that demand different standards/requirements then each client could be allocated a directory in which all the program defaults are set to suit that particular client.

Note that for the general functions common to all interactive Tribon applications a general default file named SBD_DEF1 exists, this file is described in the Tribon M3 Drafting User's Guide.

2.1 Planar Hull Modelling (sj001)

The default file for Planar Hull Modelling is called **sj001.sbd** and resides in the **SB_SHIP** directory.
The file is an ordinary text file and can be created and maintained with a standard editor. The file may contain a number of different parameters, in some cases with assigned values. If a parameter is given in the file, this means that the default action controlled by the parameter or the default value associated with the parameter is superseded. Inversely, if the parameter is not given in the default file, the default action or default value of the system is valid.

The following rules must be followed when parameters are specified in the file:

- Values assigned to parameters must be preceded by an equal sign (=).
- Commas separate multiple values.
- Parameters and their assigned values are separated by a carriage return.
- The order of parameters is irrelevant.

An extract from a sj001.sbd file is shown below:

STORE_FR
DRAW_PAN = DEFINED
SCH_CREATE

STORE_FR: When a frame number defines a value along the X-axis, the value is translated to a pure number before storing it. This means that if the frame table is then changed, the value will translate back to another frame number. To avoid this, STORE_FR can be used to actually store the frame number. This will make the panel definition follow changes in the frame table.
DRAW_PAN=DEFINED: This parameter controls the drawing of panels in a view. When set to DEFINED seams, cut-outs and notches are drawn as components. When using view-create in planar modelling the setting here will appear as the default in the menus.
SCH_CREATE: When given, and the function "Panel Store" is used, the input scheme will be created from the panel.

For a full list of all possible parameters and an explanation for each, please refer to Tribon M3- User's Guide, Tribon M3 Hull, Planar Modeling, User's Guide, Default File of Tribon Planar Hull Modelling, Parameters, General Purpose.

2.2 Curved Hull Modelling (sh700)

The default file for Curved Hull Modelling is called **sh700.sbd** and resides in the **SB_SHIP** directory.

The file is an ordinary text file and can be created and maintained with a standard editor. The file may contain a number of different parameters, in some cases with assigned values. If a parameter is given in the file, this means that the default action controlled by the parameter or the default value associated with the parameter is superseded. Inversely, if the parameter is not given in the default file, the default action or default value of the system is valid.

The following rules must be followed when parameters are specified in the file:

- Values assigned to parameters must be preceded by an equal sign (=).
- Commas separate multiple values.
- Parameters and their assigned values are separated from other parameters by carriage return.
- The order of parameters is irrelevant.

An extract from a sh700.sbd file is shown below:

BEV_LINE_SYMBOL
LP_TERM_OUT=0
SHX_PARTITION=4000

BEV_LINE_SYMBOL: When given, bevel symbols will be drawn in symbolic views in the same way as in separate generation, even if the extended bevel handling is used.

LP_TERM_OUT=0: Y and Z-co-ordinates can be described as LP-terms in system generated output. The value 0 indicates LP-terms with a possible offset will be used.

SHX_PARTITION=4000: The shell expansion view is created by development along frame curves. The density of the development curves can be controlled. The magnitude of the partition between curves is controlled via the default value (the exact positions of the development curves are selected from certain criteria within the program). If not given, 5000 is used.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Curved Modeling, User's Guides Interactive, Default Parameters of Curved Hull, Parameters, General Purpose.

2.3 Basic Design (sj700)

The default file for Basic Design is called **sj700.sbd** and resides in the **SB_SHIP** directory.

The contents of this file is restricted by the same rules previously explained for Planar and Curved Hull Modelling. For a list of all available parameters please refer to the previously mentioned chapters of the Planar and Curved Hull User's Guides.

2.4 Plate Nesting (se001)

The default information is stored as one assignment statement per row in the default file. An assignment statement consists of a keyword identifying the variable followed by an equal sign, followed by the default value. Since the default information is identified by a keyword, the order of the default information in the default file is irrelevant.

The default system is divided into two levels with a default file for each. The higher, superior level (the system manager level) consists of global default variables that may not be changed by the operator.
The inferior level consists of default variables that may be interactively changed by the operator while in the nesting application.

The superior default file must be assigned to the logical variable **SBH_NEST_DEF1** and the inferior default file to the logical variable **SBH_NEST_DEF2**.

An extract from a SBH_NEST_DEF1 file is shown below:

DIRECTION_DEF=1234567
DIRECTION_NAME1=TOP
DIRECTION_NAME2=BOTTOM
DIRECTION_NAME3=FORE
DIRECTION_NAME4=AFT
DIRECTION_NAME5=CL
DIRECTION_NAME6=PS
DIRECTION_NAME7=SB

DIRECTION_DEF: The default parameter DIRECTION_DEF indicates the directions that should be available when directions are inserted into the burning sketch. To get all directions the parameter shall have the value 1234567. If only TOP, AFT, CL and SB are to be shown then the value should be 1457. The order in which numbers are given is irrelevant.

DIRECTION_NAME1	TOP	Text associated with each direction
DIRECTION_NAME2	BOTTOM	
DIRECTION_NAME3	FORE	
DIRECTION_NAME4	AFT	
DIRECTION_NAME5	CL	
DIRECTION_NAME6	PS	
DIRECTION_NAME7	SB	

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Plate Nesting, Tribon Hull Plate Nesting, Initializations for Nesting, Defaults.

As well as the controlling default files mentioned above, a file has to be set up to define the burning machine data. Any number of burning machines can be handled. The complete file name should be assigned to the environment variable **SBH_BURNER_DATA**.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Plate Nesting, Tribon Hull Plate Nesting, Initializations for Nesting, The Burner Machine Data.

2.5 Generic Post Processor (sf001)

The Generic Post Processor (GPP) [sf001.exe] reads data files in the generic file format. These generic file are generated by the Tribon Hull Nesting application. The GPP is controlled by user-defined machine configuration data file(s), and produces output files containing NC machine instructions (ESSI or EIA format) to drive a variety of 2-axis burning machines.
To run the GPP the user must have a default data file specified and optionally a kerf data file specified.

Default data file
The Default data file contains parameters specifying the machine controller and variant, burner type, format of output, machine restrictions and fixed speeds. This file is tailored by the user to suit the particular requirements of the machine controller.
This file should be assigned to **SBH_GPP_CTRL**.
The file must appear in the directory assigned to **SB_SHIP** and the default file extension is def.
The parameter **GPP_CTRL_FILE** (in the nesting default file **SBH_NEST_DEF2**) must also point to this file. This time only the file name and extension should be given in single apostrophes the system will automatically look in the **SB_SHIP** directory for this file. If this parameter is not set then the Generic Post Processor cannot be ran interactively from within the Nesting application.

Kerf Data File
For certain combinations of controller and variant a Kerf data file is required to determine various values (e.g. Kerf pre-select offset values and burning speeds) for a given plate thickness and bevel side, angle and depth. The value(s) are retrieved from the file and in some cases interpolated from values in the file.
The user, from experience, normally creates the Kerf data file. The expected format is decided by the assignments to CONTROLLER and VARIANT in the default data file.
This file, if defined, should be assigned to **SBH_GPP_KERF**.
The default directory is **SB_SHIP** and the default file extension is def.

For a full explanation of the make-up of these two files, please refer to the Tribon M3 - User's Guides, Tribon M3 Hull, Manufacturing, Post Processors, Generic Postprocessor.

2.6 Hull PPI Programs

ⓘ *When creating views and extracting marking information for plates, all blocks are searched for intersecting panels.*
A file containing a list of blocks to be excluded can be assigned to the environmental variable SBH_EXCLUDE_BLOCKS. This allows the user to exclude blocks from these extraction and marking processes.

It should be noted that the result of running many of the Hull PPI programs is the production of drawings and list files. The drawing will by default be created in the **SBD_PDB** drawing databank. Some of these plots are usually only viewed as a check and then deleted, therefore their storage in the **SB_PDB** is undesirable.

To avoid the use of the **SBD_PDB** the variable **SBH_RECEIPT** can be set to a new database and all temporary plots will be stored here. This avoids the filling of the main drawing database and allows the regular emptying of the **SBH_RECEIPT** database to remove obsolete plots. If output from these programs is required to be stored then individual databanks can be assigned for each type of output.

The drawings output from the Hull PPI programs can be stored in databanks assigned to the following variables:

SBH_WCOG_DWG	Weight and Centre of Gravity Drawings
SBH_PARTLIST_DWG	Part List Drawings
SBH_PSKETCH_DWG	Profile Sketches
SBH_NSKETCH_DWG	Nesting Sketches
SBH_PLJIG_DWG	Plate Jig Drawings
SBH_BENDTEMPL_DWG	Bending Templates
SBH_PPART_DWG	Planar Part Drawings
SBH_CPART_DWG	Curved Part Drawings
SBH_PINJIG_DWG	Pin Jigs
SBH_MARK_PICT	Hull Marking Pictures

2.7 Plane Part Generation (sf416d)

The default file for Plane Part Generation is called **ppanparts.ip** and resides in the **SB_SHIP** directory.

This program splits plane panels into their individual plate and profile parts. The plate and profile parts will also be supplied with marking in this function, all automatically evaluated from the model.

Parameters for this program are given in an ordinary ASCII file. An extract from a ppanparts.ip file is shown below:

SHRINKAGE, SBSHRINK,
SPLIT_BEVEL_INFO,
SPLIT_EXCESS_INFO,

SHRINKAGE, SBSHRINK, With this parameter given, this program will handle the compensation for shrinkage. In this example SBSHRINK is the name of the object containing data for shrinkage compensation.
An empty string should be given if no object exists.

SPLIT_BEVEL_INFO Bevel information (normally defined for each limit of a plate, if any defined) will be split into more accurate intervals, taking the geometry of cutouts, holes, etc. into consideration.

SPLIT_BEVEL_INFO must be given if the bevel information function should work properly in the Nesting system.

SPLIT_EXCESS_INFO, Excess information (normally defined for a whole limit) will be split into more accurate intervals,taking the geometry of cutouts, holes, etc. into consideration.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3 User's Guides, Tribon M3 Hull, Manufacturing, Manufacturing of Plane Panel Parts, Plane Panel Parts, Set-up of Program, Set-up of the IP file.

2.8 Parts Lists (sf101d)

There is no designated parameter file to control the Parts List program. However a number of steps, Tribon objects, drawing forms, etc must be in place to allow the functioning of the program.
Before running the Parts List program ensure:

1. The parts to be listed have been processed through the Plane Parts Generation program. If the planar panels have not been split then nothing will appear in the parts list.

2. The object **__TB_PARTNAME_CTRL__** exists in the **SB_OGDB**. This object dictates the make-up of the resulting part names. See chapter 3.15.
Note: if the **__TB_PARTNAME_CTRL__** object is using positions numbers in the final part name then ensure position numbers have been allocated to the relevant parts.

3. As well as producing CSV lists, the system also creates a drawing for the resulting plate parts and a drawing for the resulting profile parts. For the system to produce these drawings the necessary drawing forms must exist in the **SBD_STD**
These drawing forms should be named as follows:
TB_PARTLIST_1 Drawing form for the plate parts list.
TB_PARTLIST_2 Optional drawing form for the plate parts list. If defined, then this form will be used for pages two and following pages. Drawing form TB_PARTLIST_1 will be used for page one.
TB_PROFLIST_1 Drawing form for the profile parts list.
TB_PROFLIST_2 Optional drawing form for the profile parts list. If defined, then this form will be used for pages two and following pages. Drawing form TB_PROFLIST_1 will be used for page one.

For a full list of available drawing form rules to customize the drawing forms please refer to Tribon M3 User's Guides, Tribon M3 Hull, Miscellaneous Hull Functions, Parts Lists, Parts Lists, Output Drawings, Drawing Form Rules.

4. The generated drawings mentioned above will be named automatically as follows.
PL_<six-digit number>_<page number>(<total number of pages) For plate parts.
PR_<six-digit number>_<page number>(<total number of pages) For profile parts.

The <six-digit number> is generated as specified in the file assigned to **SBH_PARTLIST_NAMES**
This file should reside in the **SB_SHIP** directory and have the following contents.
<start number> <end number> <warning number>
<current number>
An example would be as follows: **000001 001000 000990**
000069
The first drawing will be numbered 000001. The final drawing will be numbered 001000. At 000990 the system will issue a warning that available drawing numbers are running out. The last drawing produced by the system was numbered 000069

2.9 Profile Sketch and List (sf628d)

Profile sketches can be generated automatically when executing the Tribon Profile Cutting Interface or Tribon Profile Interface. These sketches are produced only if the parameter **PSKETCH** is set to **YES** in the profile restriction file. The profile restriction file should reside in the **SB_SHIP** directory and be assigned to the variable **SBH_PROF_RESTRICT**.

The customisation of profile sketches is controlled by the file assigned to the variable **SBH_SKETCH_RESTRICT**. This file should reside in the **SB_SHIP** directory. This file should be used to tailor automatic profile sketches to suit the customer's requirements.

An extract from this file is shown below:

FORM_NAME=PSKETCH_STRAIGHT
FORM_NAME_CURVED=PSKETCH_CURVED2
FORM_NAME_SHELL=PSKETCH_SHELL
AXIS_DIST=10

FORM_NAME=PSKETCH_STRAIGHT The name of the drawing form to be used for straight planar profiles. In this case the drawing form is called **PSKETCH_STRAIGHT** and this drawing form must exist in the **SBD_STD** database.

FORM_NAME_CURVED=PSKETCH_CURVED2 The name of the drawing form to be used for curved planar profiles. In this case the drawing form is called **PSKETCH_CURVED2** and this drawing form must exist in the **SBD_STD** database.

FORM_NAME_SHELL=PSKETCH_SHELL The name of the drawing form to be used for shell profiles. In this case the drawing form is called **PSKETCH_SHELL** and this drawing form must exist in the **SBD_STD** database.

*If **FORM_NAME_CURVED** and **FORM_NAME_SHELL** are not set then all sketches will use the form assigned to **FORM_NAME**.*

AXIS_DIST=10 Distance to X axis line from the base of the Profile sketch. The default is 50 mm.

For a full list of all available parameters in the **SBH_SKETCH_RESTRICT** file and a full list of available drawing form rules for the drawing forms please refer to the following document. Tribon M3 User's Guides, Tribon M3 Hull, Manufacturing, Profile Manufacturing, Production Output Profiles, Automatic Generation of Profile Sketches, Creating profile Sketches.

2.10 Weight and Centre of Gravity (sf102d)

There is no designated parameter file to control the Weight and Centre of Gravity program. However a number of steps and drawing forms must be in place to allow the functioning of the program.
Before running the WCoG program ensure:

1. The parts to be treated have been processed through the Plane Parts Generation program. If the planar panels have not been split then nothing will appear in the WCoG list.

2. As well as producing CSV lists the system also produces a drawing for each WCoG result. For the system to produce the drawing the necessary drawing forms must exist in the **SBD_STD**
   These drawing forms should be named as follows:
   TB_WCOG_1 Drawing form for the WCoG drawing.
   TB_WCOG_2 Optional drawing form for the WCoG drawing. If defined, then this form will be used for pages two and following pages. Drawing form TB_WCOG_1 will be used for page one.

   For a full list of available drawing form rules to customize the drawing forms please refer to Tribon M3 User's Guides, Tribon M3 Hull, Miscellaneous Hull Functions, WCOG Weight Calculation, Weight Calculations of Steel Structure, Output Drawings, Drawing Form Rules.

3. The generated drawings mentioned above will be named automatically as follows.
   WCOG_<six-digit number>_<page number>(<total number of pages)
   The <six-digit number> is generated as specified in the file assigned to **SBH_WCOG_NAMES**
   This file should reside in the **SB_SHIP** directory and have the following contents.

   <start number> <end number> <warning number>
   <current number>

   An example would be as follows: **000001 001000 000990**
   000069

   The first drawing will be numbered 000001. The final drawing will be numbered 001000. At 000990 the system will issue a warning that available drawing numbers are running out. The last drawing produced by the system was numbered 000069

2.11 Curved Plate Generation (sf831d)

The default file for Curved Plate Generation is called **cpanparts.ip** and resides in the **SB_SHIP** directory.
Development of plates is done automatically when definition is made at modelling stage. This program adds information like marking, shrinkage, etc. and releases the developed plate for production.
Parameters for this program are given in an ordinary ASCII file.
An extract from a cpanparts.ip file is shown below:

MARK_TEMPL,
MARK_LONG,
MARK_TRANS,
MARK_FR,
NOMINALCONTOUR,

MARK_TEMPL,
MARK_LONG,
MARK_TRANS,
MARK_FR,

These IP's define what type of objects and curves shall be used for marking on the plate. If none is given, marking will be done for plates, longitudinals and transversals. Giving any of the available IP's will automatically reset the default selection.

NOMINALCONTOUR, If this parameter is entered, then the nominal contour will be stored as a marking contour in the developed plate object. The nominal contour is the outer contour when excess and bevel are disregarded.

A comma must be used after each parameter.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3 - User's Guides, Tribon M3 Hull, Manufacturing, Automatic Generation of Curved Parts, Release of Curved Parts for Production, Set-up of Program, Default file.
As well as updating the developed plates with all requested marking information the system also produces drawings of the developed plates. To get drawings, drawing forms with fixed names **TB_CPANPARTS_1** and **TB_CPANPARTS_2** must exist on the data bank assigned to **SBD_STD**.
The drawing forms should be designed by the user to suit their own format and contain some (but maybe not all) of the available drawing form rules. The rules **$3998** and **$3999** are mandatory and are used to position the plate in the drawing form.

For a full list of available drawing form rules please refer to the Tribon M3 User's Guides, Tribon M3 Hull, Manufacturing, Automatic Generation of Curved Parts, Release of Curved Parts for Production, Set-up of Program, Set-up of drawing forms.

2.12 Bending Templates (sf820d)

The default file for Bending Templates is called **bendtempl.ip** and resides in the **SB_SHIP** directory.
The running of the bending template program results in the treated curved plates being rewritten to the **SB_PLDB** and being extended with some information about the position of the templates. The created templates are also stored in the **SB_PLDB**. The names of the templates will be the plate name extended with a running number.
An extract from a bendtempl.ip file is shown below:

DISTTOEDGE,50,
MAXDISTTOTEMPLATE,3500,
MINHEIGHTOFTEMPLATE,200,
PINS,100,
NORMAL_TEMPLATES, MANDATORY,

DISTTOEDGE,50, This parameter controls the distance between the first template and the corresponding edge of the plate. If missing it is set to 100 mm. Valid also for the positioning of the last template.
MAXDISTTOTEMPLATE,3500, Maximum allowable distance between two adjacent templates.
MINHEIGHTOFTEMPLATE,200, Minimum allowable height of a template.
PINS,100, If the parameter is given, the program will calculate and list heights for adjustable pin templates. The value given will be the distance between the pins. Default value is 200 mm
NORMAL_TEMPLATES, [MANDATORY,] This word states that the program is allowed to choose between frame templates and Normal Templates. The transition takes place at the angle 75 degree. If the parameter MANDATORY is given, the program is told to always create Normal templates.

A comma must be used at the end of each parameter line.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Bending Templates for Shell Plates, Running Environment, Control Information.

As well as updating the curved plates in the **SB_PLDB** and creating the template objects themselves, the system will also produce drawing output relating to the bending templates.

There are two ways in which the pictures of the bending templates may be output.

1. If no special set-up is made the pictures will appear in a receipt drawing in the **SBH_RECEIPT** database, therefore this database must be defined.
2. The pictures may be output on a predefined drawing form, as described below.
   Two drawings will be output:
   a) A formatted drawing will get the name **BTPL_<running_no.>_<computer name>_<job_no.>_1(2)**.
   b) An unformatted drawing with the remaining pictures will be output in a drawing with the name **BTPL_<running_no.>_<computer name>_<job_no.>_2(2)**

   Conditions to get a formatted receipt drawing are:
   A drawing form with the fixed name **TB_BENDTEMPL** must exists on the data bank assigned to **SBD_STD**. The IP **ONLY_SIMPLE_SKETCH** must not have been given in the default file.

For a full list of all possible drawing rules and an explanation for each, please refer to the Tribon M3 - User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Bending Templates for Shell Plates, Output, Drawings.

2.13 Jig Pillars (sf824d)

The default file for Jig Pillars is called **jigpillar.ip** and resides in the **SB_SHIP** directory.

This module calculates information about Jig pillars for curved shell panels. The pillars are located in the nodes of a fixed mesh i.e. their positions are predefined and their heights are calculated.

An extract from a jigpillar.ip file is shown below:

TRUE_SURFACE,
NOPILLHEIGHT,
NOSEAMPILLHEIGHT,
PILLARDISTANCE,750,

TRUE_SURFACE,	If this parameter is given, the jig calculation takes into account the actual plate thickness of a curved panel when creating the jig row curves. Input to the program must be an existing curved panel.
NOPILLHEIGHT,	If this parameter is given, the jig pillar heights will not be drawn in the jig plan sketch
NOSEAMPILLHEIGHT,	If this parameter is given, the jig pillar heights at seams will not be drawn in the jig pillar sketch.
PILLARDISTANCE,750,	Denotes the distance between the jig pillars within each row. If the parameter is not given it will be set to 1000mm by the program.

A comma must be used at the end of each parameter line.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Generation of Jig Pillars, Control Information.

As well as producing lists and creating the jig objects themselves, the system will also produce drawing output relating to the jig pillars.

There are two ways in which the pictures of the jig pillars may be output.

1. If no special set-up is made the pictures will appear in a receipt drawing in the **SBH_RECEIPT** database, therefore this database must be defined.

2. The pictures may be output on a predefined drawing form, as described below.
   Two drawings will be output, the first named **JPIL_<computer name>_<job_No>_1(2)** and the second named **JPIL_<computer name>_<job_No>_2(2)**.

   To get a formatted receipt drawing a drawing form with the fixed name **TB_JIGPILLAR** must exists on the data bank assigned to **SBD_STD**.

For a full description of the contents of the two drawings and a list of all possible drawing rules with an explanation for each, please refer to the For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Generation of Jig Pillars, Result, Drawings.

2.14 Plate Jigs (sf821d)

The default file for Plate Jigs is called **platejig.ip** and resides in the **SB_SHIP** directory.
This module generates curved jigs which, when placed on the workshop floor, provide the necessary curved space surface, to which the curved shell plates will fit. The jigs are calculated, so that they will be vertically located on the floor, and so that the panel will be as horizontal as possible or placed according to the definition of the assembly plane via input. An extract from a platejig.ip file is shown below:

ENGLISH,
MAXJIGHEIGHT, 1000,
MINJIGHEIGHT, 500,

ENGLISH,	If this parameter is given, the text on the resulting listings will be in English. This is the default language.
MAXJIGHEIGHT,	If this parameter is not given the maximum jig height will be set to 1500 mm. If a jig part is higher than the maximum jig height given it will be cut off.
MINJIGHEIGHT,	If this parameter is not given the minimum jig height will be set to 1000 mm. The distance between the floor and the panel will be set to the minimum jig height specified.

A comma must be used at the end of each parameter line.

For a full list of all possible parameters and an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Plate Jigs, Control Information.

As well as producing lists the system will also produce drawing output relating to the plate jigs. There are two drawings produced, both appear in the **SBH_RECEIPT** database, therefore this database must be defined.
One of the drawings appears in a formatted style therefore the drawing form **TB_PLATEJIGS** must exist in the **SBD_STD** database.
The drawings will be named as follows; the first named **PJIG_<pc name>_<job_No>_1(2)** and the second named **PJIG_<pc name>_<job_No>_2(2)**.

For a full description of the contents of the two drawings and a list of all possible drawing rules with an explanation for each, please refer to the Tribon M3- User's Guides, Tribon M3 Hull, Manufacturing, Curved Plates, Plate Jigs, Result, Drawings.

2.15 Profile Nesting (sf605d)

There is no designated parameter file to control the Profile Nesting program. The majority of the control over this program is the result of the file assigned to the logical **SBH_PROF_RESTRICT**. Information regarding the contents and set-up of this file can be found in the **Customizing a Project** chapter of this guide.

Although the above file controls the behavior of the majority of the Profile Nesting program, the actual arrangement/orientation of profiles within a nest is controlled by the logical **SBH_PROFILE_ROTATION**.

This logical can have one of the following values:

NO No rotation of profiles within a nest will take place. If the logical is not set then this is assumed.
UDEE Allows up-down and end-end rotation of the profiles
UD Allows only up-down rotation of the profiles
EE Allows only end-end rotation of the profiles

Obviously not all rotation options within a profile nest are relevant to all profile types. The table below shows what is allowed for each profile type.

Profile Type	NO	UD	EE	UDEE
10	1	1	1	1
20	1	0	0	0
21	1	0	0	0
30	1	0	0	0
31	1	0	0	0
33	1	0	0	0
35	1	0	0	0
36	1	0	0	0
37	1	0	0	0
38	1	0	0	0
40	1	0	1	0
43	1	0	1	0
50	1	0	1	0
51	1	1	1	1
52	1	1	1	1
53	1	1	1	1
54	1	1	1	1
55	1	1	1	1
56	1	1	0	0
60	1	1	1	1
61	1	0	1	0
62	1	0	1	0
63	1	1	1	1
64	1	1	1	1
65	1	1	1	1
70	1	1	1	1
71	1	1	1	1
72	1	1	1	1
73	1	1	1	1
74	1	1	1	1

A **1** in the column indicates that the rotation option is applicable to that particular profile type.

A **0** in the column indicates that the rotation option is not applicable to that particular profile type.

Note:
If the logical is set to **UDEE**, i.e. maximum rotation, the system will only apply the rotations to the profile types as per the table shown.

For example, even with **UDEE** set the system will never rotate bulb bar (type 20) profiles within a profile nest.

3 Customising a project

3.1 Cutouts and Clips

In shipbuilding it is usual that the stiffening of the main structural members penetrate the plating of subordinate elements, e.g. so that longitudinal frames in the shell or in the decks/bulkheads pass through webs and double floor bottoms. Such penetrations are called **cutouts** in Tribon. Cutouts often need to be reinforced by small plate pieces called **clips** or collars in Tribon. The clips are always defined in association with the cutouts but they will be discussed separately in this document.

3.1.1 Cutouts

Cutouts are normally standardised regarding general shape, radii, clearances, etc. and have well defined dependencies on profile types and sizes. Therefore they lend themselves well to be generated by different types of parameterised macros.

There are two options for generation of cutouts that are currently available in Tribon.
These options are:

1. The majority of cutouts used by a certain yard are similar to those used by all other yards in their general principles. They may vary in details regarding e.g. clearances, radii, etc. but the pattern is common. For this category of cutouts Tribon has an "External Cutout Definition Facility" which allows a customer to set up his own standards.

2. Very special and yard specific cutouts with an arbitrary geometry can be built up using Tribon geometry macros which allows a customer to develop cutouts with any shape.

Independently of the way they have been created all types of cutout can be used both in plates and profiles. Cutouts in Tribon are identified and picked by a number that can be selected quite arbitrarily by the customer when setting up the cutout standard.

3.1.2 Tribon External Cutout Definition Facility

The Tribon External Cutout Definition Facility contains "templates" for a number of typical cutouts, some of which are used by virtually all shipyards. Each customer can create his own "instances" of these cutouts with radii, clearances, etc. in accordance with his needs and practices.

The "templates" have reserved numbers but new versions can be made using arbitrarily selected user numbers.

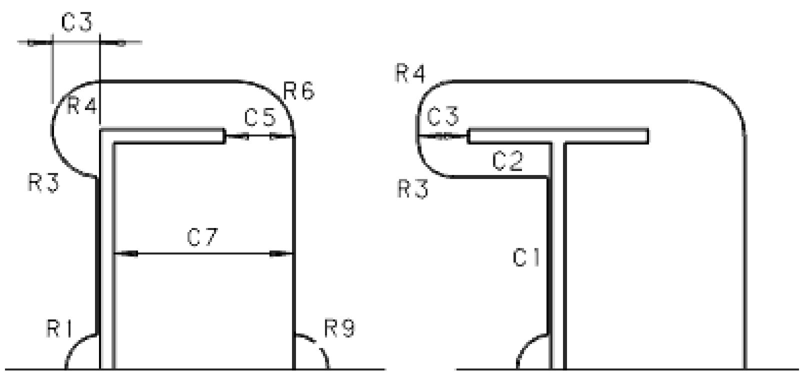
The cutouts will take into account the fact that the angle between web and flange may not always be 90 degrees depending on the orientation of the penetrating profile relative to the plate.

Cutouts in the external cutout definition facility are described in a text file in the general TIL language.
The definition file may be given an arbitrary name. This file is used to create an object named **__CUTSTDOBJ__** in the **SB_OGDB**. This object contains the customer cutout definitions and is referenced by the applications during the modelling stage.
An extract from an external cutout definition file is shown below:

```
CUTOUT, 308
/TYPE=10
/PROF=31

/ARC=1 /RAD=15
/ARC=3 /RAD=15
/ARC=4 /RAD=(30, 40)
      /H_LIM=220
/ARC=6 /REF=4

/GAP=1 /DIST=1.5
/GAP=3 /REF=3 /DIST=1.5
/GAP=4 /DIST=40
/GAP=5 /DIST=25
;
```



CUTOUT: Is the number by which a user picks a certain cutout. There may be several occurrences of the same number but then the profile types must be different.

TYPE: Specifies the main type of which the cutout is an occurrence. Must be picked from one of the basic types available.

PROF: Specifies the profile types for which the current cutout should be applicable.

ARC=1: Defines the arc at R1 as RAD=15. This results in a 15mm radius.

ARC=3: Defines the arc at R3 as RAD=15. This results in a 15mm radius.

ARC=4: Defines the arc at R4 as RAD=(30,40). This results in the option of either a 30mm or 40mm radius.

H_LIM: This value is used to decide which of the radii defined in ARC=4 should be used. For profiles with height up to and including 220mm a 30mm radius will be applied. For profiles with height in excess of 220mm a 40mm radius will be applied.

ARC=6: Defines the arc at R6 as REF=4. This results in the same arc criteria as already defined for ARC=4 being applied at ARC=6.

GAP=1: Defines the clearance at C1 as 1.5mm.

GAP=3: Defines the clearance at C3 as REF=3/DIST=1.5. This results in a clearance of 15mm (the value assigned to ARC=3) plus an additional gap of 1.5mm, therefore a total clearance of 16.5mm

GAP=4: Defines the clearance at C4 as 40mm.

GAP=5: Defines the clearance at C5 as 25mm.

For a detailed description of the available Tribon “templates” and the TIL file used to define customer “instances” please refer to the Tribon Documentation, Tribon M3-User’s Guide, Set-up and Customisation, Cutouts and Clips , External Profile Cutout Definition.

After creating/updating the file start the **Initiate Hull Standard** program.

Go to the **Cutouts and Clips** section and click on the **Cutouts, ext. def., create** option.

Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.1.3 Cutouts via Macros

Before a cutout can be defined by a macro the customer must create the cutout macro file. A description and example of a cutout macro can be found in the Tribon Documentation, Tribon M3 Set-up and Customising → Cutouts and Clips → Profile Cutouts via Macros.
When a cutout macro has been defined, it has to be compiled before it can be called from the Tribon system. This may be done using the geomac command. The location of the source, list and resulting binary files are controlled by three Tribon environment variables:

SBB_GEO_MACRO_SRC
SBB_GEO_MACRO_LST
SBB_GEO_MACRO_BIN

Make sure that these variables have an appropriate definition and then issue the command "geomac" to perform the interpretation. Observe that the command does not run but only performs an interpretation of the macro.
The connection between a customer controlled cutout definition macro and its cutout code by which it can be accessed from within e.g. Tribon Hull Modelling is defined in an ASCII file. Its full file specification should be given by the Tribon environment variable **SBH_CUTOUT_MACRO**. If this variable is undefined, Tribon will simply ignore the handling of cutouts via macro. For information regarding the make up of this ASCII file please refer to the Tribon M3-User’s Guide, Tribon M3 Hull, Set-up and Customisation, Cutouts and Clips → Profile Cutouts via Macros

3.1.4 Automatic setting of Cutouts

When designing the internal structure in Tribon the cutouts for penetrating profiles are supposed to be selected from a set of available cutout types. This document describes an alternative facility that lets the customer define default cutouts. A cutout is default for a specific combination of a profile type and a panel data type. The facility can be used to simplify the establishment of cutouts in the Planar Hull Modelling module.

The definition of default cutouts takes place via an ordinary text file, created and maintained in the standard editor of the computer system. The name of the file can be selected quite freely and its total file specification should be assigned to the Tribon environment variable **SBH_CUTOUT_CTRL**. This file is normally supposed to be stored in the default directory of the current project.

The syntax of the file is described below. Please note that everything that is written in the same line in the syntax description must be so. Maximum line width is 132 characters.

TYPE = <proftype_1>
<panel_dt> <defcut> <defclip>
<panel_dt> <defcut> <defclip>
...
TYPE = <proftype_2>
<panel_dt> <defcut> <defclip>
<panel_dt> <defcut> <defclip>

<proftype_i> One of the normal profile types used in Tribon, e.g. 10, 20 and 30.

<panel_dt> Data type of the current panel. This data type together with the profile type is the key to find the appropriate default cutout. <panel_dt> may contain the "wild card" character "%" in one or several positions.
Examples are "410", "%2%", "%%%" and "%12". In this document <panel_dt> will also be referred to as a panel data type mask.
One panel data type mask can be more specific than another. A panel data mask, A, is less specific than another mask, B, if A contains more "wild card" characters than B. Example: "12%" is less specific than "121".
The system **does not** check for multiple or overlapping panel data type masks. If two masks overlap each other then the first one will be applied. Because of this the most specific data type masks should appear first in the definition file.

<defcut> Cutout type to be applied to the current profile type if there is a match of the data type of the current panel.

<defclip> Clip arrangement that should be applied. Zero indicates no clips.

An extract from a cutout control file is shown below:

TYPE=10

410	**38**	**011**
41%	**42**	**0**
4%%	**1**	**0**
1%%	**511**	**011**
%%%	**202**	**004**

After creating the file and assigning it to the logical **SBH_CUTOUT_CTRL**, reset the project and start the **Initiate Hull Standard** program.
Go to the **Cutout and Clips** section and click on the **Autom. cutout sel., check** option.
Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.1.5 Cutout Setting set up file.

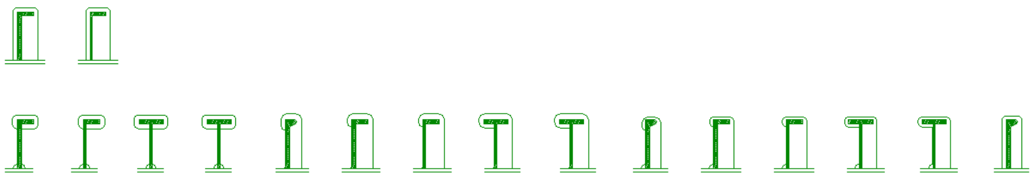
The name of the set-up file may be chosen arbitrarily and is given as input to the inithull function for set-up of Cutout Setting. However, the file extension should be .dat.
The result of the interpretation (if successful) will be a "Cutout Standard Drawing" stored in the SB_SETTINGS databank named "__SBH_CUTOUT_TYPES__". Moreover, a list file will be produced during the process of interpretation of this file. Inithull is normally run via the Tribon Job Launcher and the list file will be stored as defined in the set-up for the Job Launcher.

An example of the file is shown below:

Cutout	Profile
101	30
101	31
101	40
101	43
308	20
308	30
308	31
308	40
308	43
309	20
309	30
309	31
309	40
309	43
311	20
311	30
311	31

The file contains the cutout number or name followed by a space, then the profile type to which it is applicable.

An example of the output drawing held in the SB_SETTINGS databank is shown below:



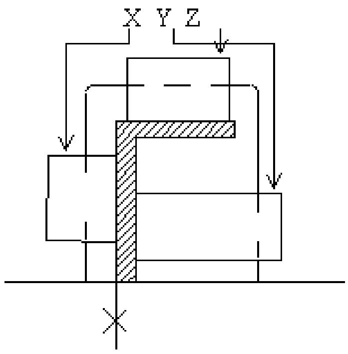
In the example shown, when using Planar Hull, if cutouts are to be added to a type 20 profile, only cutouts type 308,309 & 311 would be available from the selection dialogue box.

3.1.6 Clips

In shipbuilding it is most common that the stiffening of the main structural members penetrate the plating of subordinate elements. Such penetrations are in Tribon called **cutouts**. Normally the penetrated plate is connected to the penetrating stiffener, either for strength reason or simply to tighten the penetration (or a combination of both). Often the cutout itself produces a part of this connection; part of it may be welded to the profile. However, very often the connection must be completed by extra plate pieces. The parts used for fastening/sealing are always referred to as **clips** in Tribon, although it sometimes would be more relevant to call them collars or lugs.

Within Tribon a range of standard clips are delivered. These clips are shown opposite:

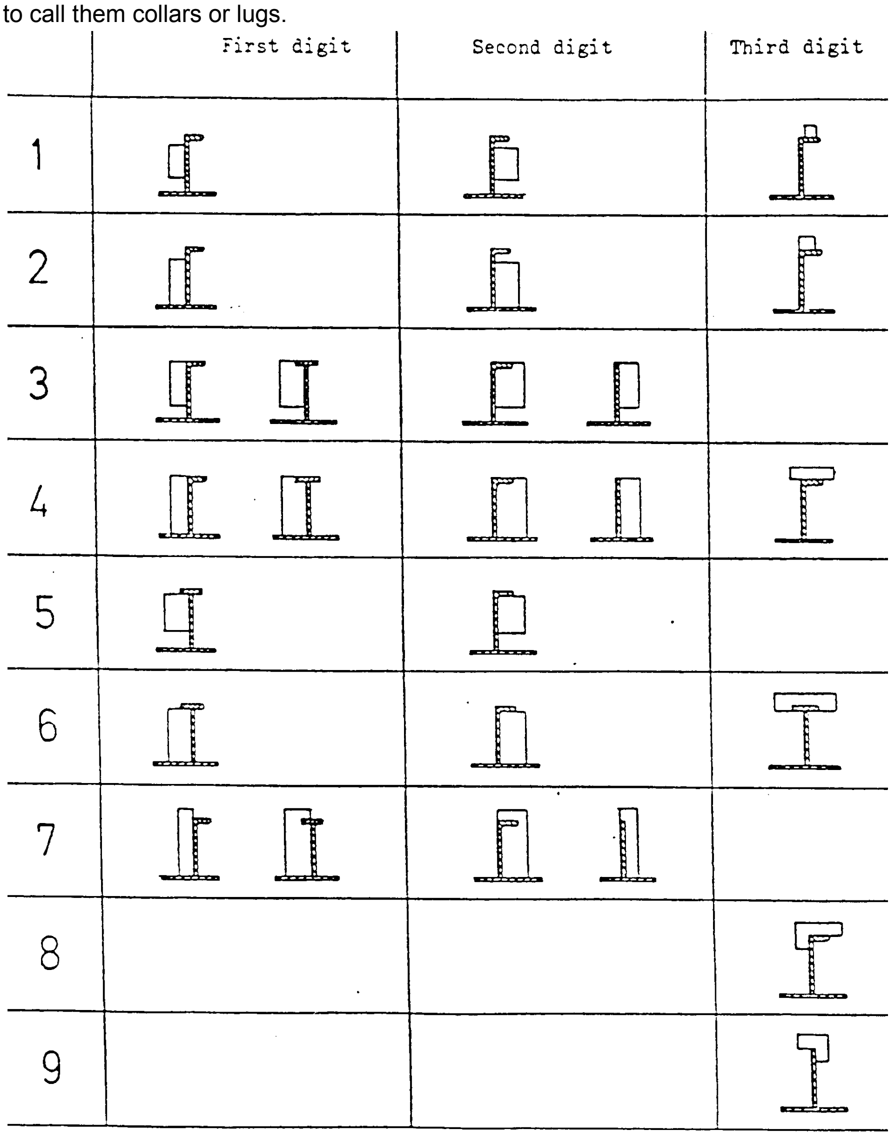
The addition of clips, to a previously defined cutout statement, is achieved by the inclusion of a valid 3-digit clip code. If we assume the three digits are represented by X, Y and Z

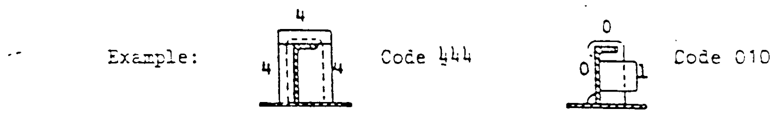


X – Valid for clip on mould line of profile
Y – Valid for clip on non-mould line of profile
Z – Valid for clip on the top of the profile

As well as defining the position of the clips, the number allocated to X, Y and Z also specifies the type of clip to be fitted in accordance with the standards shown. If clips are to be omitted at any position this must be indicated by a 0 (zero).

Note: It is also possible to generate clips via macro's. See Tribon M3-User's Guide, Tribon M3 Hull, Set-up and customisation, Cutouts and Clips , Clips via Geometry Macros.





3.2 Profiles

3.2.1 Profiles for projects

Within the Tribon system a large number of profile types and parameters are available for use. However, for a certain project, it may be desirable to make only a subset of profile types available when interactively modelling. There are two methods available to achieve this, the first method described has been used in earlier versions of Tribon and is still valid if __SBH_PROF_TYPES__ drawing does not exist in the SB_SETTINGS databank.

The restriction of the available profiles takes place via an ordinary text file, created and maintained in the standard editor of the computer system. The name of this file can be selected quite freely and its full file specification should be assigned to the Tribon environment variable **SBH_PROF_CTRL**. The file is normally stored in the default directory of the current project.

Method 1

The layout of this file is described by the example below. The format is free but everything that appears in the same line in the example must be so. Maximum line width is 132 characters.

```
TYPE = 10 'Flat bar'
100, 10
100, 12
120, 12
150, 12
200, 25
250, 25
TYPE = 20 'Bulb bar'
200, 9
220, 10
240, 10
260, 12
TYPE = 30 'L-bar'
150, 90, 10
200, 100, 10
250, 120, 12
350, 150, 12
```

The keyword **TYPE** is the profile type as defined in the Tribon standard. The text on the same line as the **TYPE** keyword is the customer description of the profile type, i.e. it can be translated or modified to suit your standard terminology. This text will appear on a menu when creating stiffeners interactively. The lines following the **TYPE** keyword select a number of dimensions for the current profile type. The parameters should be given in the same order as they are entered in modelling input. There is no restriction as to the number of lines selecting profile dimensions.

After creating the file and assigning it to the logical **SBH_PROF_CTRL**, reset the project and start the **Initiate Hull Standard** program.

Go to the **Profiles and Flanges** section and click on the **Profiles for project, check** option.

Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

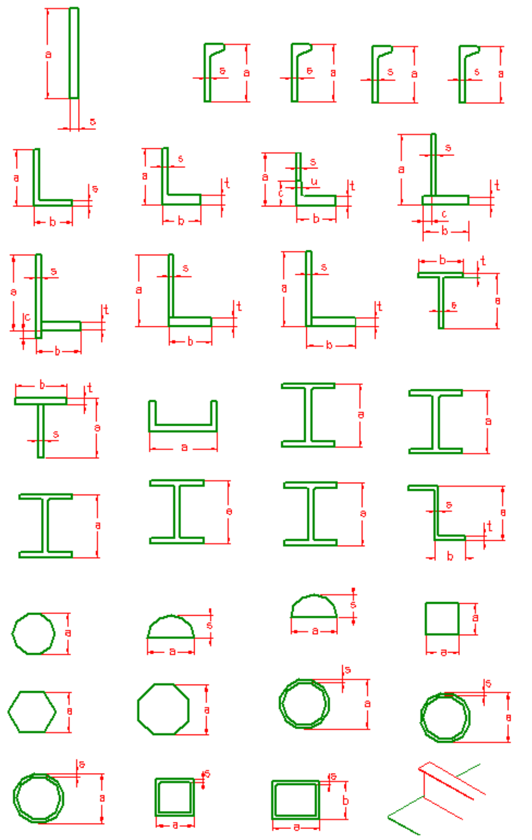
Method 2

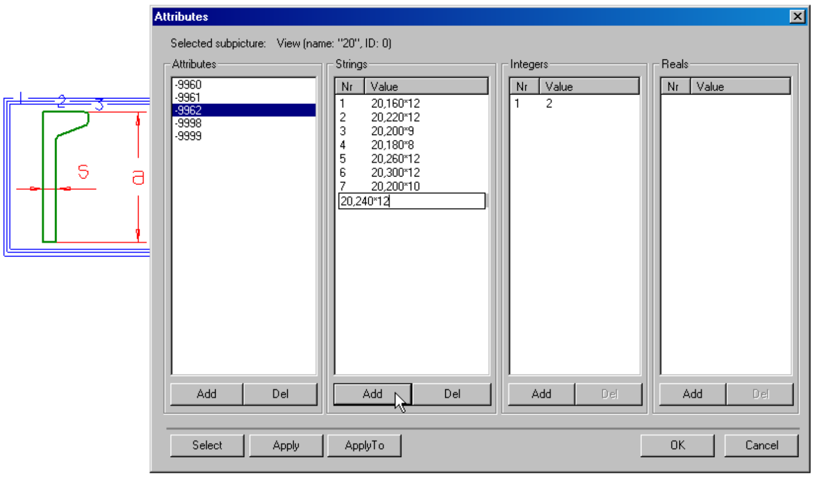
A drawing named __SBH_PROF_TYPES__ should exist in the SB_SETTINGS databank (a copy can be transferred from the Tribon template project if none exists in the current project, the drawing will appear as shown opposite:

Each view is created as a separate sub-picture with attributes that define the available profile sizes.

Select **Tools > Inspect Drawing** and select option **10 Attributes**

Select the sub-picture of the profile you wish to create or modify instances of, the sub-picture will be highlighted, select level 1. The following menu will be displayed.





From the list of attributes select **9962**, The Strings field will display the current values available for the selected profile. Select Add and key the profile type and size to be added e.g. 20,240*12, when finished select **OK** and save the drawing. The new profiles will now be available for selection.

3.2.2 U and I bar set-up

Tribon has a set of predefined and reserved profile types. Certain profile types e.g. I-bars and U-bars are picked by giving only nominal sizes. This traditionally involves defining a nominal height and sometimes, when several different profiles have equal height, the flange width. Tribon has a number of built-in tables for these profile types dimensions according to DIN-standards.

However the facility exists that allows a customer to set up his own tables to replace the defaults for these profile types, i.e. for the profile types 50 through 59. It is possible to redefine any of the profile types and still use the defaults for others.

The definition of the profiles takes place via an ordinary text file, created and maintained in the standard editor of the computer system. The name of the file can be selected quite freely and its full name should be assigned to the Tribon environment variable **SBH_I_U_BARS**.

The file is divided into a number of statements. Each statement must be written in a separate line and the maximum line width is 80 characters. Spaces between terms are of no importance, but no blank lines are allowed. Note that each statement must be terminated by a semicolon (**;**).

TYPE=<proftype>, 'comment';
[ANGLE=<fla_angle>;]
<H>, <B>, <Tw>, <Tf>, <R1> [, <R2>];
<H>, ... ;

TYPE: This statement indicates the start of a definition of a certain profile type. `<proftype>` can be any type of 50, 51, 52, 53, 54, 55 and 59. The text given as a comment after the **TYPE** statement is the customer description of the profile type and is not used by Tribon.

ANGLE: This statement sets the flange inclination of the lower inner surface of the flange and will be valid for all following profiles until the next **ANGLE** or **TYPE** statement. Default value after an **ANGLE** statement is **0**. This statement is optional and may be omitted as well as given once for every line of profile data.

The **parameter statements** have no statement keyword but contain only the parameters controlling the shape of the profile section.

H: The height of the profile. This is the actual height of the profile and not the nominal height (if different).

B: Width of the profile.

Tw: Thickness of the web.

Tf: Thickness of the flanges.

R1: Radius between web and flange.

R2: Second radius (interpretation varies between profiles.

The parameter lines following the **TYPE** statement specify a number of occurrences of this profile type. There are no restrictions as to the number of lines specifying profile dimensions i.e. any number of profiles may be defined.
The measures may be given in imperial units according to TRIBON standards.
The lines need not be sorted on dimensions.
The types 50, 51 and 59 must have six parameters, while 52, 53, 54 and 55 have only five.

An example of the input file is shown below:

TYPE=51,'I_BAR';
80,70,4.0,6.0,3,1;
150,60,3.0,5.0,3,1;

After creating the file and assigning it to the logical **SBH_I_U_BARS**, reset the project and start the **Initiate Hull Standard** program.

Go to the **Profiles and Flanges** section and click on the **U-and I-bar set-up, check** option.

Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

When using Method 1 to define profile types, these should also be included U & I bars should also be included on the SBH_PROF_CTRL file to make them available for selection in the modelling applications.
When using Method 2 to define profile types, follow the instructions shown in section 3.2.1.

3.3 Connection codes

When generating the hull model it is recommended to use topological references as much as possible. This is valid also for profiles since a profile is usually generated by connection of the ends to another profile or to a surface.

The detailed nature of the end connection of profiles is controlled by two separate factors: the **connection code** and the **endcut code**. Moreover, the endcut code and the connection code in combination control the selection of symbols at the ends of stiffeners in drawings.

The **connection code** defines the position of the end of the profile relative to the component to which it is connected, e.g. butting/overlap/clearance and possibly offsets of the mould lines of involved profiles.

Tribon Hull has an in-built set of standard connection codes. This chapter describes how a customer can create a set of connection codes of his own to be used in parallel with, or replacing, the in-built standards. The definition of connection codes takes place via an ordinary text file, created and maintained in the standard editor of the computer system. The name of a connection code file can be selected quite freely and its full file specification should be assigned to the Tribon environment variable **SBH_CONCODES**.

The file is a free format text file where the definition of each connection code requires four numbers.
It is recommended to define one connection code in one line. The numbers should be separated by at least one blank.

Formally, the layout may be described in the following way.

<user code> <type> <dist> <offset>

<user code> Is a number in the range 1-9999 that should be used by the designer. It must be unique within the file but may very well be equal to one of the default codes. In that case the standard code is redefined.

<type> Defines the main type of connection. A picture of each type of connection can be found in the Tribon User's Guide – Tribon M3 Set-up and Customising – Profiles in Tribon – Connection Code Definition – Connection Control File – Layout of the Connection File.

<dist> Distance between the profile end and the component to which it is connected.

<offset> Offset of the planes of the profiles. Relevant only for **<type>**=1.
>0 Offset in the direction of the material (flange) of the profile to which the connection takes place.
<0 In the opposite direction.

An example of part of a connection code definition file can be seen below:

```
1   1  100   0
2   1   75   0
3   1   50   0
4   1    0   0
5   1    0   5
6   1    0   8
7   1    0  10
8   1    0  12
9   1    0  15
10  2  100   0
11  2   75   0
12  2   50   0
13  2  -50   0
14  3   50   0
15  3    0   0
16  3   75   0
17  3  100   0
18  3  150   0
19  3   40   0
20  2 -100   0
```

After creating the file and assigning it to the logical **SBH_CONCODES**, reset the project and start the **Initiate Hull Standard** program.

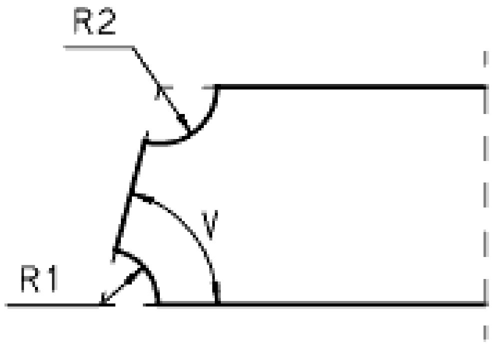
Go to the **Profiles and Flanges** section and click on the **Connection codes, check** option.
Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.4 Endcut set-up

When a profile is manufactured, e.g. by cutting of a raw bar, the ends of it must normally be prepared to get a suitable shape corresponding to its intended use. In Tribon the shape of the end of profiles is called the ***endcut*** of the profile.

Tribon contains a vast in-built endcut standard that is delivered to all customers. The standard is divided into a number of predefined ***endcut types***. An endcut type corresponds to a certain geometrical pattern. Its actual shape may be controlled by a number of parameters, some of which are set up in a standard table. Others must be given by the designer or are calculated automatically by Tribon, e.g. depending on the connection in which the profile end is involved. A typical example of a simple endcut for a flat bar is illustrated below (Tribon standard endcut type 11).

Three different types of data items control the shape of this and all other endcuts.



1. The endcut type (in the example above the endcut type is 11).
2. A number of implicitly defined parameters with standard values. The parameters that may have standard values are individual to a certain endcut type. These parameters have a restricted number of values and each combination of them is identified by an ***endcut code***. The standardised parameters are either radii or fixed angles of the endcut and they cannot be given explicitly by the designer but are selected via the endcut code. In the figure above **R1** and **R2** are such parameters.
3. A number of additional parameters which may take any value and which are explicitly given by the designer or automatically calculated by Tribon. In the example above **V** is such a parameter. These parameters are given as explicit complements to the endcut code.

For a more detailed explanation of endcut types, endcut codes and explicit endcut parameters please refer to the Tribon M3 User's Guide – Tribon M3 Set-up and Customising – Profiles in Tribon – Profile Endcuts in Tribon – Endcut Standards in Tribon, Principles.

For a view of the full set of endcuts currently delivered with the system please refer to the Tribon M3 User's Guide – Tribon M3 Hull, Tribon M3 Set-up and Customising, Profiles in Tribon, Endcut Standards, Survey of Endcut Types.

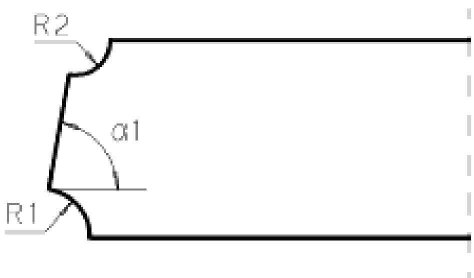
3.4.1 Endcut table

The customer can modify the in-built parameters of the endcut standard via an endcut table. The geometry of any endcut can be controlled by a restricted number of parameters. The interpretation of these parameters is individual for each endcut type. The parameters are **A**, **B**, **C**, **R1**, **R2**, **V1**, **V2**, **V3** and **V4**.

Code, parameter	R1	R2
1100, α1	0	0
1102, α1	0	35
1104, α1	0	50
1105, α1	0	60
1106, α1	0	75
1110, α1	35	0
1112, α1	35	35
1114, α1	35	50
1115, α1	35	60
1116, α1	35	75
1120, α1	50	0
1122, α1	50	35
1124, α1	50	50
1125, α1	50	60
1126, α1	50	75
1130, α1	75	0
1132, α1	75	35
1134, α1	75	50
1135, α1	75	60
1136, α1	75	75
1140, α1	60	0
1142, α1	60	35
1144, α1	60	50
1145, α1	60	60
1146, α1	60	75
1190, α1	KS10	0

A, **B** and **C** are lengths, **R1** and **R2** normally notch radii and **V1**, **V2**, **V3** and **V4** angles of the endcut.
Depending on the endcut type some of the parameters become irrelevant.

The figure below shows the in-built type 11 endcuts for flat bars:



The extract below shows the portion of the endcut table set-up file that deals with the previously displayed type 11 endcuts for flat bars.

```
600 '__ENDCUTTAB__'
601    11   9   -1   -1   -1   1   2   100   -1   -1   -1
601 1100  3   0.0000000001   0.0000000001   -1
601 1102  3   0.0000000001   35            -1
601 1104  3   0.0000000001   50            -1
601 1105  3   0.0000000001   60            -1
601 1106  3   0.0000000001   75            -1
601 1110  3   35         0.0000000001   -1
601 1112  3   35         35            -1
601 1114  3   35         50            -1
601 1115  3   35         60            -1
601 1116  3   35         75            -1
601 1120  3   50         0.0000000001   -1
601 1122  3   50         35            -1
601 1124  3   50         50            -1
601 1125  3   50         60            -1
601 1126  3   50         75            -1
601 1130  3   75         0.0000000001   -1
601 1132  3   75         35            -1
601 1134  3   75         50            -1
601 1135  3   75         60            -1
601 1136  3   75         75            -1
601 1140  3   60         0.0000000001   -1
601 1142  3   60         35            -1
601 1144  3   60         50            -1
601 1145  3   60         60            -1
601 1146  3   60         75            -1
601 1190  3   -10        0.0000000001   -1
```

name of endcut object
selection row for type 11
parameter rows defining endcut codes for type 11 endcuts

The first line of the endcut table must always be a record type 600, which is used to define the resulting name of the endcut table. In the example above the endcut table will appear as **__ENDCUTTAB__** in the **SB_OGDB**. Therefore in the project file the variable **SB_ECUT** must be set to the resulting name of the endcut table. This record type 600 must appear as the first line of the endcut table file and must appear only once in the file.
The next line in the endcut table file is the selection row for endcut type 11. A selection row must appear once for each type of endcut, these selection rows are created by Tribon Solutions and should not be altered by the customer.
The remaining lines in the example are parameter rows defining the relevant parameter values for each endcut code for type 11 endcuts. Each line should begin with a record type 601, then the resulting endcut code, then a digit specifying the number of parameters to follow, then the relevant number of parameters.

It is at this parameter row level that the user can customise the endcuts available in Tribon. This can be done by editing an existing line or adding a completely new line with a completely new code. Currently the following rules should be adhered to when creating new endcut codes. For endcut types **21**, **22**, **26**, **27**, **28**, **31**, **32**, **35** and **36** endcut codes ending in **0** should always have a default flange angle 90 degrees, those not ending in **0** should have a flange angle with a different value. After creating/updating the file and assigning it to the logical **SB_ECUT**, reset the project and start the **Initiate Hull Standard** program.
Go to the **Profiles and Flanges** section and click on the **Endcut table, create** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.4.2 Endcut selection

The definition of default endcuts takes place via an ordinary text file, created and maintained in the standard editor of the computer system. This file allows the user to define a subset of relevant endcuts from the total set available according to TRIBON standards. It is also possible to define default endcuts to be picked automatically depending on the type of connection. An endcut may in this way be selected for a specific combination of profile type and connection code.

```
TYPE = 11 `User defined string´
PROF = 10
1102
1100
1104,30 CON=5,10,12
1110,40

TYPE = 12 `User defined string´
PROF = 10
1210,80 CON= 81
1210,100 CON= 13
1210,13
```

The name of the endcut file can be selected quite freely and its total file specification should be assigned to the TRIBON environment variable **SBH_ENDCUT_CTRL**. The file is normally supposed to be stored in the default directory of the current project. The layout of the file is described by an example opposite. The format is free but everything that appears in the same line in the example must be so. Maximum line width is 132 characters.

The keyword **TYPE** is the endcut type as defined in the standards. This keyword starts a "type group" and may occur any number of times. The text given on the same line as the **TYPE** keyword is the customer description of the endcut group.

The keyword **PROF** defines for what profiles types this endcut group is relevant. Any number of profile types can be given but there must be a minimum of one. The lines following the **PROF** keyword select a set of endcut codes. Each of these lines starts with an endcut code according to TRIBON standards. It is also possible to set predefined values for the "free" parameters of the endcut (optional).

With the keyword **CON** it is possible to define a number of connection codes. Doing this means that the endcut on the current line is the default for the profile type(s) in combination with the connection code(s). Maximum number of connection codes following the **CON** keyword is 20. Use of the keyword **CON** is optional. Note that the same endcut may appear several times, e.g. with different parameters, to be the default for different connections (cf. **1210** in the second "type group" in the example).

The input example above should be interpreted in the following way: when generating a flat bar and using any of the connection codes **5**, **10** or **12** the default endcut type for that end is **1104** with parameter **30**. However, the designer may select any other endcut at will.

After creating the file and assigning it to the logical **SBH_ENDCUT_CTRL**, reset the project and start the **Initiate Hull Standard** program. Go to the **Profiles and Flanges** section and click on the **Endcut selection, check** option. Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.5 Profile restriction file

The Profile Nesting Utility of TRIBON allows the customer to specify the working limits for a profile robot. If these limitations have been set up, the profile nesting application has a function to check if any profile nesting contains parts that violate them. If so, this nesting is stored with a signal that it should be fabricated manually.

However, it is possible to get information in the modelling phase about the profiles that do not comply with the restrictions set up by the customer. The user defines these restrictions in a normal ASCII-file assigned to the logical variable **SBH_PROF_RESTRICT**. The file is keyword oriented and can by handled by any editor.
The contents of the file are divided into one general information part, valid for all types of profiles and one part that is type dependent. All profile types to be used in a project should have their own corresponding section in this file.

3.5.1 General information section

Some examples from the general section of the profile restriction file can be seen below. Please note this is not a complete list only an extract to highlight certain keywords.

CHECK_CURVMOD=YES
CHECK_HULLMOD=YES
COMMON_CUT=NO
PSKETCH=YES
NSKETCH=YES
MANSTIFF=YES
MANSTIFF_CSV=YES
MIN_DIST=5
USE_ENDCUT=YES

CHECK_CURVMOD **CHECK_HULLMOD**	If set to **YES** there is a warning given from the relevant modelling application if the profile created is outside defined restrictions.
COMMON_CUT:	Information to the system whether common cut between profiles is allowed or not.
PSKETCH:	Must be set to **YES** if sketch output is required when running the Profile Cutting Interface.
NSKETCH:	Must be set to **YES** if sketch output for nested profiles is required when running the Profile Cutting Interface
MANSTIFF:	Must be set to **YES** if manufacturing list output is required when running the Profile Cutting Interface
MANSTIFF_CSV:	Must be set to **YES** if manufacturing list output in comma separated format is required when running the Profile Cutting Interface
MIN_DIST:	Minimum distance in mm between profiles or between profile and raw material edge.
USE_ENDCUT:	Information to the system whether the scrap material produced when cutting an endcut shall be used for nesting.

For a full list of all applicable keywords in the general section of the file, please refer to the Tribon M3 User's Guide, Tribon M3 Hull, Set-up and Customisation, Setup for Production, Set-up for Profile Fabrication in Tribon, The Restriction File, Keywords in the General Section of the Restriction File.

3.5.2 Type specific section

As mentioned previously, each profile that is to be used in a project must have a section of the profile restriction file dedicated to it. Some examples from the type specific section of the profile restriction file can be seen below. Please note this is not a complete list only an extract to highlight certain keywords.

PROF_TYPE=10
MIN_RAW=1000
MAX_RAW=16000
MIN_PROF=500
MAX_PROF=16000
MAX_WEIGHT_M=250
TEXT_HEIGHT=20
TEXT_WIDTH=20
WANTED_SCRAP=0
TRUE_SCRAP=900
OVERLENGTH=500
MIN_DIM1=80
MIN_DIM2=6
MAX_DIM1=550
MAX_DIM2=38
ENDCUT=11
BEVEL=20

PROF_TYPE:	The number used is according to the design standard of TRIBON, e.g. 10 for flat bars, 20 for HP bars, etc. The section is ended by a new **PROF_TYPE** keyword or by end of file.
MIN_RAW:	Minimum length in mm of raw material that can be handled by the robot. Raw materials below this length will be handled manually.
MAX_RAW:	Maximum length in mm of raw material that can be handled by the robot.
MIN_PROF:	Minimum length in mm of profiles that can be handled by the robot.
MAX_PROF:	Maximum length in mm of profiles that can be handled by the robot.
MAX_WEIGHT_M:	Maximum weight/metre of profile to be handled by robot.
TEXT_HEIGHT:	Height of text characters signed on profiles by robot.
TEXT_WIDTH:	Width of text characters signed on profiles by robot.
WANTED_SCRAP:	Size of any wanted scrap in mm at end of each raw material.
TRUE_SCRAP:	Maximum length in mm of scrap possible to handle by the robot used. Normally it's the basin width in the robot.
OVERLENGTH:	Overlength in mm to be added to all profiles to be bent.
MIN_DIM1:	Minimum size of parameter one of current profile type to be handled by the robot.
MIN_DIM2:	Minimum size of parameter two of current profile type to be handled by the robot.
MAX_DIM1:	Maximum size of parameter one of current profile type to be handled by the robot.
MAX_DIM2:	Maximum size of parameter two of current profile type to be handled by the robot.
ENDCUT:	Endcuts allowed by the robot shall be defined with this keyword. The keyword can be used any number of times.
BEVEL	Bevel types possible to be handled by a robot are defined with this keyword. The keyword can be used any number of times.

For a full list of all applicable keywords in the profile type dependant section of the file, please refer to the Tribon M3 User's Guide, Tribon M3 Hull, Set-up and Customisation, Setup for Production, Set-up for Profile Fabrication in Tribon ,The Restriction File, Keywords in the Profile Section.

After creating the file and assigning it to the logical **SBH_PROF_RESTRICT**, reset the project and start the **Initiate Hull Standard** program.
Go to the **Production Support** section and click on the **Profile restrictions, check** option. Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.5.3 Shell stiffener curvature

A curved shell profile is normally bent in the workshop before it is mounted in the ship. If the profile is only slightly curved this step may not be necessary, the profile can be manufactured as a straight one and will then "fall into place" when mounted in the ship.
The facility described here makes it possible for the yard to set up rules to decide if a shell stiffener should be stored as curved or straight on the profile data bank, **SBH_PROFDB**. The rules are applied during this storing process.
These rules are defined via a text file whose full path and name should be assigned to the logical name **SBH_SHELLPROF_BENDING_CTRL**. The file is organised in statements, written in TIL-format.

Each statement in the text file has the following syntax:

PROF/TYPE=<profile type> **/ HEIGHT=**<profile height> **/ METHOD=**<method> **/ CVAL=**<contol value>

PROF/TYPE: The profile type for which **METHOD** and **CVAL** are valid. If this keyword is omitted then the values in this statement will be used as defaults for all profile types.

HEIGHT: The biggest height of the particular profile type for which **METHOD** and **CVAL** are valid.

METHOD: The method used to evaluate if a shell profile is curved. Valid choices are **WHOLE_TRACE_INDEP_DIST** and **PER_1000_FACT**. The two methods are described below.

CVAL: Described below for each **METHOD**.

Method 1: **WHOLE_TRACE_INDEP_DIST**

WHOLE_TRACE_INDEP_DIST stands for "whole trace independent distance". With this method the largest distance is measured between a line between the end points of the profile and the trace. If this distance is bigger than **CVAL** the profile is considered as curved.

Thus, if **d > CVAL** then the profile is considered as curved.

Method 2: **PER_1000_FACT**

When the method **PER_1000_FACT** is used a number of chords (Li) with equal length (=1000) are computed along the curve. For each of the chords the maximum distance Xi is calculated.

Let the biggest of these Xi values be called X. If **(CVAL*<profile length>) > X** then the profile is considered to be straight
An example of the control file for profile type 10 may appear as shown below:
PROF /TYPE=10 /HEIGHT=100 /METHOD=PER_1000_FACT /CVAL=0.0025;
PROF /TYPE=10 /HEIGHT=200 /METHOD=PER_1000_FACT /CVAL=0.0018;
PROF /TYPE=10 /HEIGHT=10000 /METHOD= PER_1000_FACT /CVAL =0.0005;
PROF /HEIGHT=10000 /METHOD= WHOLE_TRACE_INDEP_DIST /CVAL=5;
The first line handles profile type 10 with heights up to 100 mm.
The second line is valid for profile type 10 with heights between 100 and 200 mm.
The third line for profile type 10 has a height that is greater than any possible profile height.

The last line is the default for profile types not listed in the file. This is in fact what you get if **SBH_SHELLPROF_BENDING_CTRL** is not defined.

For a full explanation of this setting, please refer to the TRIBON M3 User's Guide – TRIBON M3 Set-up and Customising – Set-up for Production – Set-up for Profile Fabrication in TRIBON – Curvature control of shell stiffeners.

After creating the file and assigning it to the logical **SBH_SHELLPROF_BENDING_CTRL**, reset the project and start the **Initiate Hull Standard** program. Go to the **Production Support** section and click on the **Shell stiff. curvature, check** option. Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.5.4 Raw profiles

If the profile nesting option is to be used a file must be in place to provide the system with a list of available raw profile sizes. When the profile nesting system is ran the user can select a specific raw profile or the system can read the complete file and nest the profiles into the most suitable raw profile.

This file defining the available raw profiles should be assigned to the environment variable **SBH_RAW_PROFILES** and should reside in the **SB_SHIP** directory.

An extract from a raw profile definition file is shown below:

```
'FB06008' 10 2 60 8 0 12000 'A' 500
'FB10008' 10 2 100 8 0 12000 'B' 500
'FB10010' 10 2 100 10 0 12000 'C' 500
'FB10012' 10 2 100 12 0 12000 'D' 500
'FB10015' 10 2 100 15 0 12000 'E' 500
'FB10018' 10 2 100 18 0 12000 'F' 500
'FB10025' 10 2 100 25 0 12000 'G' 500
```

The make up of the file should be as follows:

<ident> <prof type> <parameters> <par1> <par2> <parN> <length> <buying mark> <no. off>

<ident> The raw profile designation. An identification code for the raw profile. It has to be unique within the current run. It is a string of characters with the maximum length of 26 characters.

<prof type> The raw profile type. A code number defining the profile type according to the Tribon standards. It is an integer.

<parameters> The number of profile parameters. It depends on the profile type. It is an integer.

<par1>, etc The profile parameters according to the Tribon standards. The number of parameters is given by the <parameters> value.

<quality code> The raw profile steel quality. A code number designating the profile steel quality. It is an integer. (See 3.12 Material Qualities, int_code).

<length> The length of the raw profile.

<buying mark> The buying mark of the raw profile. It is a string of characters with the maximum length of 26 characters.

<no.off> The number of available raw profiles in the current run. It is not updated during or after each run.

3.6 Brackets

Brackets are often standardised regarding their general shape, how they are stiffened and not least how they are related to other elements that they connect or support. For this type of standardised brackets TRIBON has an advanced facility that allows a customer to set up his own brackets standards.

A customer's bracket definition is kept in an object named **__SBH_BRACKET_CTRL__**, which is stored in the **SB_OGDB**. This object is created via a text file, which should be assigned to the variable **SBH_BRACKET_CTRL** in the project definition file. Within this file the customer can stipulate which brackets should appear as interactive options when a user selects one of the bracket syntaxes.

The file **template_kcs_bkt_ctrl.dat**, which can be found in the **\M3\customise** directory, contains a large range of example bracket definitions. For a full description of all the statements in this file please refer to the Tribon M3 User's Guide, Tribon M3 Hull, Setup and Customisation, Brackets.
Below is an example that could be used to create bracket types **KL** and **KLK**.

```
FLANGE,FLA_1_1
        /TYPE=1
        /END1_FLANGE=100
        /END2_FLANGE=100
        /H=(100, 160, 220, 280)
        /RULE=1
        ;
TOE,T1
        /TYPE=1
        /TOE_H=15
        ;
CON,C1
        /TYPE=1
        /CL_TS=50
        /CL_OS=-50
        /CL_FREE=20
        ;
BRACKET, KL
        /SYNTAX=(4,2,1,3,7,8)
        /SYMBOL=26
        /TOE_A=T1
        /CON_A=C1
        /TOE_B=T1
        /CON_B=C1
        /NOTCH="
        ;
BRACKET, KLK
        /SYNTAX=(4,2,1,3,7,8)
        /SYMBOL=27
        /TOE_A=T1
        /CON_A=C1
        /TOE_B=T1
        /CON_B=C1
        /FLANGE=FLA_1_1
        /NOTCH="
        ;
```

FLANGE: FLA_1_1: A name (string) that uniquely identifies a certain set of heights and calculation rules (<= 8 characters)

TYPE: Number that specifies the flange type to be used in this set. Must be selected from one of the types defined in the **__SBH_FLANGE_CTRL__** object.

END1_FLANGE: Number that specifies the flange end type to be used at end one of the flange (close to arm A of the bracket). Must be selected from one of the flange end types defined in the **__SBH_FLANGE_CTRL__** object.

END2_FLANGE: Number that specifies the flange end type to be used at end two of the flange (close to arm B of the bracket). Must be selected from one of the flange end types defined in the **__SBH_FLANGE_CTRL__** object.

H: Specifies allowable flange heights. The heights should be sorted in increasing order.

RULE: The flange height is selected based on the length of the free edge (= **L**), divided by 10. I.e. the flange height should be picked from the set of heights, assigned to **H** based on its relation to **L/10**.

= 1 Select closest height to L/10
= 2 Select closest to L/10 but smaller
= 3 Select closest to L/10 but larger

Creation of the actual geometry of the flange is controlled by the **__SBH_FLANGE_CTRL__** object.

TOE: **TYPE:** Basic toe type, compulsory. Must be picked from the available toe types documented in the TRIBON M3 User's Guide.

TOE_H: Toe height

CON: **C1:** A unique user defined name of the connection type. Its length is restricted to eight (8) characters.

TYPE: Registered basic connection type with allowed values. For a full description of all available connection types please refer to the TRIBON M3 User's Guide – TRIBON M3 Set-up and Customising – Brackets – Customer Set-up of Brackets – Statement Types, Detailed Specification – CONNECTION Statement.

BRACKET: **KL:** Is the unique name of the bracket, used when the bracket is generated. The name should start (and preferably end) in a letter and be maximum 8 characters long.

SYNTAX: Specifies the connection cases (syntaxes) in which the bracket may occur. The first of the given syntaxes is the default syntax, i.e. if not explicitly specified by the user this syntax will be supposed in the generation of the bracket. Syntax number 8 is the "free syntax" when all bracket parameters will have to be supplied by the user.

SYMBOL: Specifies the symbol number within symbol font 91 which is used in the interactive picking of the bracket type when generating the bracket. If not given it is not possible to pick the bracket by a symbol of it. It is the responsibility of the customer to create an appropriate symbol for the bracket.

TOE, CON NOTCH, FLANGE: Specify

As can be seen from the previous example the bracket definition file is built up by a number of statements. The statements may be of different types, each statement type used for a specific task. They are organised in a hierarchical way so that certain statement types create information that is used in the "higher level" statements. Typically, the low level statement types define standard bracket toes, standard connections of bracket toes and available profile types. The very top-level statement type is the one that defines a bracket with reference to much of the information, created by the other statements. The statements must be given in a logical order, i.e. information that is referred to from one statement must have been defined in another statement earlier in the input file.

After creating/updating the file and assigning it to the logical **SBH_BRACKET_CTRL**, reset the project and start the **Initiate Hull Standard** program. Go to the **Other Standard Set-up** section and click on the **Brackets std, create** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.6.1 1.8.2 Bracket Instance Object, Create

The bracket set-up facility of Tribon allows a customer to define a bracket standard of his own. In doing that some parameters of the brackets are given specified values, e.g. toe heights, toe lengths, etc. whereas other parameters are supposed to be evaluated or given by the user when the bracket is used.
This section specifies in detail a facility by which a customer may define default values for some bracket parameters other than those that can be defined in the normal bracket set-up file. Examples of such parameters are those defining the size and thickness of a bracket. A set of bracket parameters for a certain bracket type is said to form an *instance* of the bracket type.
Bracket instance definitions are created by writing definition data in an ordinary text file in a special language based on the Tribon Interpretative Language (TIL). The format and the handling of the bracket instance definition file are described in detail below.
The definition file may be given an arbitrary name. In order to make the bracket instance definitions available to the application programs it should be compiled by a function of the hull utility *inithull*. If the compilation is successful a bracket definition object will be stored in the structure data bank (associated with **SB_OGDB**) by name **__SBH_BRACKET_INSTANCE__**.
The compilation will also result in a receipt list with input and errors, if any. If *inithull* has been started via the Tribon Job Launcher this list can be accessed like all other files resulting from a Job Launcher run. If *inithull* has been started outside the Job Launcher the receipt list will be stored in the print directory of the current project with the file extension *.lst* appended to the name of the input file.
The INSTANCE statement defines the default values of a set of bracket parameters forming a bracket instance. The bracket instance is valid for the bracket type defined in a preceding BRACKET statement. The name of the bracket instance must for identification reasons start by the name of the bracket type, followed by digits or slash-characters (/).
The keywords for the bracket modelling parameters are with few exceptions the same as in the input data language of Hull Modelling.
The example below shows a bracket instance definition file for brackets B, BC & KLK.

```
BRACKET, 'B';
INSTANCE, 'B10' /A=200 /B=200 /MAT=8 /NOTCH='KS10';
INSTANCE, 'B11' /A=250 /B=250 /MAT=9 /NOTCH='KS10';
INSTANCE, 'B12' /A=300 /B=300 /MAT=9 /NOTCH='KS10';
INSTANCE, 'B13' /A=350 /B=350 /MAT=9 /NOTCH='KS10';
INSTANCE, 'B14' /A=400 /B=400 /MAT=10;
INSTANCE, 'B15' /A=450 /B=450 /MAT=10;
INSTANCE, 'B16' /A=500 /B=450 /MAT=10;

BRACKET, 'BC';
INSTANCE, 'BC10' /B=250 /MAT=8;
INSTANCE, 'BC11' /B=250 /MAT=10;
INSTANCE, 'BC20' /B=300 /MAT=10;
INSTANCE, 'BC21' /B=300 /MAT=10 /NOTCH='R50' /NOA='R50';
INSTANCE, 'BC22' /B=350 /MAT=10 /NOTCH='VU100*50' /NOA='KS10';
INSTANCE, 'BC30' /B=500 /MAT=8;

BRACKET, 'KLK';
INSTANCE, 'KLK10' /A=500 /MAT=12 /QUA='A36';
INSTANCE, 'KLK20' /A=700 /B=500 /MAT=10;
INSTANCE, 'KLK21' /A=700 /B=700 /MAT=10 /QUA='E';
INSTANCE, 'KLK22' /A=700 /B=700 /MAT=10 /QUA='A32' /NOTCH='R50';
INSTANCE, 'KLK30' /A=1100 /B=1245 /MAT=15 /QUA='A' /NOTCH='R75';
```

Definition of an instance:

<instance_name> The name of the bracket instance.
Note: that it must start with the name of the bracket type followed by digits or slash-characters (/). The bracket instance name must be unique.

A A measure.

B B measure.

C C measure.

D D measure.

H Flange height.

R Radius of the free side.

OFF Distance the bracket is displaced from the profile plane.

MAT Bracket material thickness.

QUAL Steel quality of the bracket.

BEV Bevel along connected edges of the bracket. Default value for max 5 bevel types may be defined.

NOTCH Default notch definition for use in the origin corner of the bracket.

NOA Ditto for notches at the end of arm A,

NOB at the end of arm B and

NOC at the start of arm A for a three-edge bracket.

COM A free comment, maximum 26 characters long, describing the bracket instance. Used only for identification purposes within the instance file.

AS1 General purpose string 1.

AS2 General purpose string 2.

AS3 General purpose string 3.

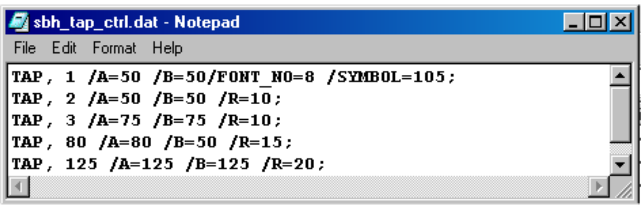
AS4 General purpose string 4.

POS Position number.

After creating/updating the file, reset the project and start the **Initiate Hull Standard** program. Go to the **Other Standard Set-up** section and click on the **Brackets instance object, create** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

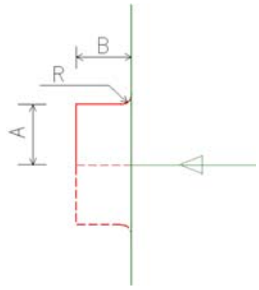
3.7 Tap Pieces

When welding plates together, there is a tendency for the edges of the plates to become damaged where the weld starts and finishes. To alleviate the problem, tap pieces can be added which can later be removed. These may be added at the junction of a seam with a plate edge or corner, inside an opening, at a cut out or notch.


sbh_tap_ctrl.dat - Notepad
File Edit Format Help

```
TAP, 1 /A=50 /B=50/FONT_NO=8 /SYMBOL=105;
TAP, 2 /A=50 /B=50 /R=10;
TAP, 3 /A=75 /B=75 /R=10;
TAP, 80 /A=80 /B=50 /R=15;
TAP, 125 /A=125 /B=125 /R=20;
```

The control file (which must have a .dat extension) contains one line describing each tap type. Dimensions A, B and R may be defined as shown in the sketch. A font and symbol number can be assigned to represent the tap in the drawing.



After creating/updating the file, reset the project and start the **Initiate Hull Standard** program. Go to the **Other Standard Set-up** section and click on the **Tap std, create** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.8 Bevel

The Tribon system comes complete with a built in range of bevels. For details of the available bevels please refer to Tribon M3-User's Guide, Tribon M3 Hull, Set-up and Customisation, Bevel Excess and Weld, Bevel Handling in Tribon,

The standard bevels are used to define an edge preparation for a plate. If any bevel gaps are to be taken into account when splitting the plate parts for manufacturing, these must be defined by the use of left and right excess during the seam creation.

If a customer wishes to define their own bevels and have the bevel gap automatically applied during plate splitting process then the Extended Bevel Handling option is required.Weld Planning Setup
There are number of possibilities to customize the result of the weld calculations. Currently the following parameters are implemented in the Weld Planning application.

User-defined Weld Positions, Weld Leg Lengths, and Weld defaults

3.8.1 Weld Positions

The user-defined weld positions are based on the Japanese Industrial Standard Z 3003-1963, *Definition of Welding Position*. In this standard is specified how to calculate the rotation angle and inclination angle for a weld, described in the local co-ordinate system of the assembly. With these angles it is possible to set up rules for calculating the fundamental weld positions.
The input file for weld positions is an ordinary text file in a simple language based on the general TIL format. This language has only one statement (POSITION) that can be repeated any number of times. The order in the input file is, however, important since the system checks the rules sequentially until a weld position is found.

The file name is weldPosition.def and must be placed in the SB_SHIP directory.

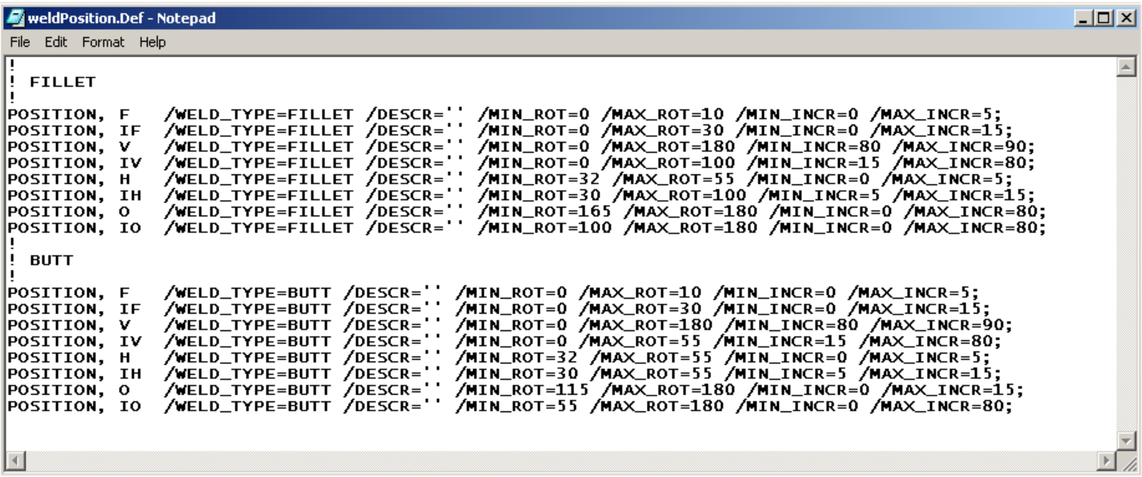
The input should follow the following syntax:

```
POSITION, <pos name> /WELD_TYPE=<weld type>
                     /DESCRIPTION=<user description>
                     /MIN_ROT=<minimum rotation angle>
                     /MAX_ROT=<maximum rotation angle>
                     /MIN_INCR=<minimum inclination angle>
                     /MAX_INCR=<maximum inclination angle>;
```

<pos name> is the string which will be displayed for the weld position, maximum 26 characters.
<weld type> is the type of weld. Possible values are 'fillet' and 'butt'.

The attributes MIN_ROT, MAX_ROT and MIN_INCR, MAX_INCR defines the limiting intervals for the rotation angle and inclination angle, respectively

Example:



```
weldPosition.Def - Notepad
File  Edit  Format  Help
!
!  FILLET
!
POSITION, F   /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=0 /MAX_ROT=10 /MIN_INCR=0 /MAX_INCR=5;
POSITION, IF  /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=0 /MAX_ROT=30 /MIN_INCR=0 /MAX_INCR=15;
POSITION, V   /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=0 /MAX_ROT=180 /MIN_INCR=80 /MAX_INCR=90;
POSITION, IV  /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=0 /MAX_ROT=100 /MIN_INCR=15 /MAX_INCR=80;
POSITION, H   /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=32 /MAX_ROT=55 /MIN_INCR=0 /MAX_INCR=5;
POSITION, IH  /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=30 /MAX_ROT=100 /MIN_INCR=5 /MAX_INCR=15;
POSITION, O   /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=165 /MAX_ROT=180 /MIN_INCR=0 /MAX_INCR=80;
POSITION, IO  /WELD_TYPE=FILLET /DESCR='' /MIN_ROT=100 /MAX_ROT=180 /MIN_INCR=0 /MAX_INCR=80;
!
!  BUTT
!
POSITION, F   /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=0 /MAX_ROT=10 /MIN_INCR=0 /MAX_INCR=5;
POSITION, IF  /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=0 /MAX_ROT=30 /MIN_INCR=0 /MAX_INCR=15;
POSITION, V   /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=0 /MAX_ROT=180 /MIN_INCR=80 /MAX_INCR=90;
POSITION, IV  /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=0 /MAX_ROT=55 /MIN_INCR=15 /MAX_INCR=80;
POSITION, H   /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=32 /MAX_ROT=55 /MIN_INCR=0 /MAX_INCR=5;
POSITION, IH  /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=30 /MAX_ROT=55 /MIN_INCR=5 /MAX_INCR=15;
POSITION, O   /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=115 /MAX_ROT=180 /MIN_INCR=0 /MAX_INCR=15;
POSITION, IO  /WELD_TYPE=BUTT /DESCR='' /MIN_ROT=55 /MAX_ROT=180 /MIN_INCR=0 /MAX_INCR=80;
```

3.8.2 Weld Leg Length

The Weld Leg Length can be calculated in two different ways. It can either be done using an input file where some simple rules are defined, or by using Execution Units.

The input file for weld leg lengths is an ordinary text file in a simple language based on the general TIL format. This language has only one statement (LEGLENGTH) that can be repeated any number of times.
The file name is weldLegLength.def and must be placed in the SB_SHIP directory.
The input should follow the following syntax:
LEGLENGTH, <leg length> /WELD_TYPE=<weld type>

/PART1_THICK=<thickness 1>
/PART2_THICK=<thickness 2>
/PART1_QUAL=<quality 1>
/PART2_QUAL=<quality 2>;

<leg length>	is the resulting leg length.
<weld type>	is the type of weld. Possible values are 'fillet' and 'butt'
<thickness 1>	is the thickness for the first part. If -1 is given no check will be made on thickness
<thickness 2>	is the thickness for the second part. If -1 is given no check will be made on thickness
<quality 1>	is the quality for the first part. If '*' is given no check will be made on quality
<quality 2>	is the quality for the second part. If '*' is given no check will be made on quality

Example:

weldLegLength.Def - Notepad

File Edit Format Help

```
!
! FILLET
!
LEGLENGTH, 3 /WELD_TYPE=FILLET /PART1_THICK= 5 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 4 /WELD_TYPE=FILLET /PART1_THICK= 6 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 5 /WELD_TYPE=FILLET /PART1_THICK= 8 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 6 /WELD_TYPE=FILLET /PART1_THICK=11 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 7 /WELD_TYPE=FILLET /PART1_THICK=14 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 8 /WELD_TYPE=FILLET /PART1_THICK=17 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 9 /WELD_TYPE=FILLET /PART1_THICK=21 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH,10 /WELD_TYPE=FILLET /PART1_THICK=25 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH,11 /WELD_TYPE=FILLET /PART1_THICK=40 /PART2_THICK=-1
                               /PART1_QUAL='*' /PART2_QUAL='*';
!
! BUTT
!
LEGLENGTH, 3 /WELD_TYPE=BUTT /PART1_THICK= 5 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 4 /WELD_TYPE=BUTT /PART1_THICK= 6 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 5 /WELD_TYPE=BUTT /PART1_THICK= 8 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 6 /WELD_TYPE=BUTT /PART1_THICK=11 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 7 /WELD_TYPE=BUTT /PART1_THICK=14 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 8 /WELD_TYPE=BUTT /PART1_THICK=17 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH, 9 /WELD_TYPE=BUTT /PART1_THICK=21 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH,10 /WELD_TYPE=BUTT /PART1_THICK=25 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';

LEGLENGTH,11 /WELD_TYPE=BUTT /PART1_THICK=40 /PART2_THICK=-1
                             /PART1_QUAL='*' /PART2_QUAL='*';
```

3.8.3 Weld Defaults

The input file for weld defaults is an ordinary text file in a simple language based on the general TIL format. This language has one statement for each default parameter. The order in the input file is irrelevant.
The file name is weldDefaults.def and must be placed in the SB_SHIP directory.
The default parameters should have the following syntax:

MINIMUM_WELD_LENGTH, <min weld length>;	All welds shorter than <min weld length> will be deleted.
WELD_TYPE_TOLERANCE, <weld tolerance>;	If the angle between to plates is less than <weld tolerance> the weld type will be butt weld, otherwise fillet weld.
CALCULATE_LEGLENGTH, <Yes/No>;	The leg length will be calculated if this default is set to *Yes*.
CALCULATE_POSITION, <Yes/No>;	The position will be calculated if this default is set to *Yes*.
CALCULATE_SUSPENSION, <Yes/No>;	The suspension will be calculated if this default is set to *Yes*.
REMOVE_FULL_SUSPENSION, <Yes/No>;	Welds with full suspension will be removed if this default is set to *Yes*.
CREATE_WELD_REPORT, <Yes/no>;	The weld report will be automatically created if this default is set to Yes.

The following default parameters can optionally be used in the creation of robot control files.

AUTHORIZATION, <authorization>;	is an arbitrary text enclosed in ''.
ORGANIZATION, <organization>;	is an arbitrary text enclosed in ''.
AUTHOR, <author>;	is an arbitrary text enclosed in ''.

Example of weld default file:

weldDefaults.Def - Notepad

File Edit Format Help

```
MINIMUM_WELD_LENGTH, 20.0;
CALCULATE_LEGLENGTH, Yes;
CALCULATE_POSITION, Yes;
CALCULATE_SUSPENSION, No;
REMOVE_FULL_SUSPENSION, No;
WELD_TYPE_TOLERANCE, 4;
CREATE WELD REPORT. Yes:
```

3.9 Shrinkage

Each welding operation (and heating in general) in metal structures causes deformation (shrinkage) of the structural elements involved. This means that if e.g. plate parts are described to their nominal sizes they will in the fabrication and assembly process become too small to fit. Thus when fabricating the parts they should already have been compensated for the shrinkage so that the nominal dimensions are reached when the assembly process is completed. TRIBON Hull has a facility for shrinkage compensation that allows a user to describe his model to nominal dimensions but to extract parts that are expanded for the shrinkage.

The experiences of shrinkage vary considerably from shipyard to shipyard. Therefore it is necessary for each customer to define their own shrinkage parameters to be used. These parameters are described in a shrinkage object, which should be stored in the **SB_OGDB**. The name of this object can be selected freely but the file used to create it should have a dat extension. The name of the shrinkage object must be given as an input parameter or default parameter in programs where the shrinkage should be applied.
After creating the shrinkage object the shrinkage compensation can be applied automatically, semi-automatically or manually to a panel during the modelling stage. Any actual adjustment to the panel's dimensions will take place during the panel splitting process, ensuring all compensations are considered before the plate reaches the nesting system.

The shrinkage can be calculated using two different methods:

1. Statistical method (standard with the hull system)
   The compensation is calculated in a primary and secondary direction. The number of profile traces on the panel is calculated and an allowance for each trace is allocated. This allowance is added as a block figure to the edges of the panel.

2. Advanced Shrinkage Handling (see the Advanced Hull Features chapter).
   The compensation is calculated in a primary and secondary direction. The compensation for each profile is applied locally in way of the trace rather than being summated and added to the edge of the panel. This leads to a higher degree of accuracy with marking lines after the compensation has been applied.

For a full explanation of the two methods of shrinkage calculation please refer to the Tribon M3-User's Guide, Tribon M3 Hull, Set-up and Customisation, Shrinkage Compensation, Shrinkage Compensation.in Tribon.

Regardless of the shrinkage method being applied the input file used to create the shrinkage object is identical. An example of the input file may appear as shown below:

```
SHRINKAGE,'__SHRINKAGE__';
FILLET/UPPER_LIM=12/COMP=1/
        UPPER_LIM=22/COMP=2;
BUTT/UPPER_LIM=12/COMP=2/
      UPPER_LIM=22/COMP=4;
LONGITUDINAL/PARTITION=500
      /UPPER_LIM=10/COMP=3/ UPPER_LIM=20/COMP=3
      /PARTITION=1500
      /UPPER_LIM=10/COMP=4/UPPER_LIM=20/COMP=2;
```

SHRINKAGE: The name of the resulting shrinkage object.

FILLET:
UPPER_LIM: The plate thickness up to which the associated shrinkage compensation should be applied. The limit itself will be included in the interval.
COMP: Shrinkage compensation in mm for thickness up to **UPPER_LIM**.

BUTT:
UPPER_LIM: The plate thickness up to which the associated shrinkage compensation should be applied. The limit itself will be included in the interval.
COMP: Shrinkage compensation in mm for thickness up to **UPPER_LIM**.

LONGITUDINAL/PARTITION: The average partition between traces for which a certain table should be used. Tables for at most 25 individual partitions may be created.
UPPER_LIM , **COMP** are used as in the **FILLET/BUTT** statement and are valid for the table associated with the closest preceding **PARTITION**.

After creating/updating the file start the **Initiate Hull Standard** program. Go to the **Other Standard Set-up** section, and click on the **Shrinkage obj., create** option.

Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.10 Swedging

Swedging is formed, by letting a tool press the plate against a counter-part (a dye). Thus the extension of the plate will become smaller in the fabrication process. Correspondingly, the size of the plate before swedging must be compensated for the swedging. This is done automatically when extracting parts from the hull model in the plane parts generation module. The centreline of the swedging is in this process added to the plate as a marking line in a similar way as for ordinary stiffeners.
The characteristics of the swedging may be very different between different yards. A certain yard may also use multiple types of swedging simultaneously. Therefore it is necessary that each customer should be allowed to control all characteristics of the type of swedging used by him.
All the characteristics of a customer's swedging are kept in a **swedging object,** which should be stored in the **SB_OGDB**. The name of this object can be selected freely but it must be given as an input parameter or default parameter in modules where the swedging is used, e.g. in Planar Hull Modelling, PPI Hull, Plane Part Generation, etc.
The swedging object is created via an ASCII file with input written in a language based on the general TIL format for interpretative input. The name of this file can be selected arbitrarily but should have a file extension dat. The input format has three different statement types and an example is shown below:

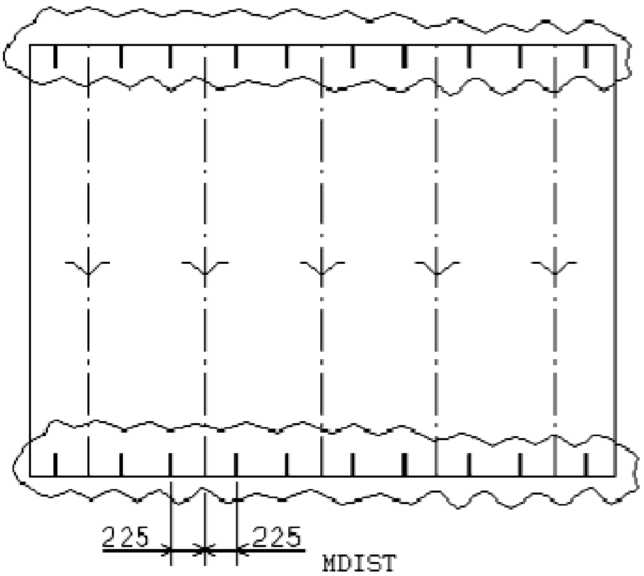
```
SWEDGE, TTPSWEDGE;
PROFILE, 100 /VIEWING = 101 /PLANE_SYMBOL = 3 /CROSS_SYMBOL = 3
/MDIST=80 /MSIZE=150 /MGAP=20;
SIZE,75
        /WIDTH = 90
        /R1 = 30
        /R2 = 30
        /LOWER_LIM = 5
        /UPPER_LIM= 10 /COMP = 68.1
        /UPPER_LIM= 20 /COMP = 69.2
        /UPPER_LIM= 30 /COMP = 70.2
;
SIZE,100
        /WIDTH = 120
        /R1 = 30
        /R2 = 30
        /LOWER_LIM = 5
        /UPPER_LIM= 10 /COMP = 78.1
        /UPPER_LIM= 20 /COMP = 79.2
        /UPPER_LIM= 30 /COMP = 80.2
;
PROFILE, 101 /VIEWING = 103 /PLANE_SYMBOL = 1 /CROSS_SYMBOL = 1
/MDIST=80 /MSIZE=150 /MGAP=20;
SIZE,75
        /WIDTH = 90
        /UPPER_WIDTH = 30
        /R1 = 30
        /R2 = 30
        /LOWER_LIM = 5
        /UPPER_LIM= 10 /COMP = 58.1
        /UPPER_LIM= 20 /COMP = 59.2
        /UPPER_LIM= 30 /COMP = 60.2
;
```

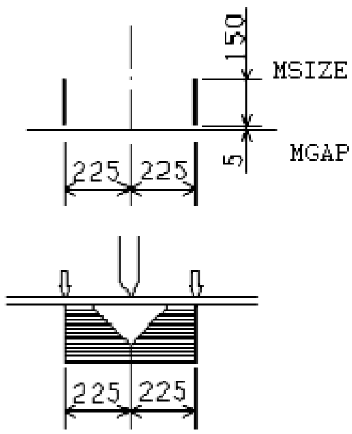
SWEDGE: Specifies the name of the swedging object.

PROFILE: Specifies a certain type of swedging by a number (100-110 only are valid) to be used when modelling swedged panels. It also defines certain parameters controlling the views of swedging, e.g. line type and symbols to be displayed in drawings.

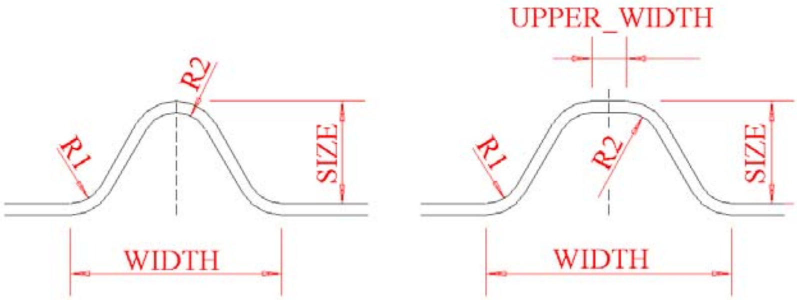
MDIST, MSIZE, MGAP: Parameters controlling marking lines (See sketch below for clarification).

SIZE: A certain "profile type" may occur in several predefined sizes (nominal heights). This statement type specifies a certain profile size and some data related to that size. (See sketch on page 41 for further details).





```
SIZE,75
        /WIDTH = 90
        /UPPER_WIDTH = 30
        /R1 = 30
        /R2 = 30
        /LOWER_LIM = 5
        /UPPER_LIM= 10 /COMP = 58.1
        /UPPER_LIM= 20 /COMP = 59.2
        /UPPER_LIM= 30 /COMP = 60.2
;
```



LOWER_LIM defines the lower plate thickness for the current range, UPPER_LIM defines the upper plate thickness for the current range (for any subsequent ranges this will be interpreted as the LOWER_LIM). COMP defines the expanded allowance to be added to the plate length for each swedge.

For a full explanation of the three different statement types please refer to the Tribon M3-User's Guide, Tribon M3 Hull, Set-up and Customisation, Knuckled Pieces and Swedging, Swedging (Small Corrugations) in Tribon, User Set-up of Swedging Characteristics.

After creating/updating the file start the **Initiate Hull Standard** program. Go to the **Knuckled Pieces** section and click on the **Swedging, create** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.11 Knuckled panel bending control

During the modelling of a knuckled panel a series of sub-panels are defined and the system is informed that these sub-panels should be connected to form a main knuckled panel. However when this main knuckled panel reaches the production stage it is actually one plate and it has to be bent to the desired knuckled shape. This knuckling of the plate is achieved by the use of a bending tool. The characteristics of the bending tool, like radius and stretching compensations, may be set up in a knuckle control file.
The file is an ordinary ASCII text file and its full file specification must be assigned to the TRIBON environment variable **SBH_KNUCKLE_CTRL**. The file is built up in the following way:

- It must always start with a **TYPE** statement defining the connection type code, e.g. TYPE=45 (this is the type of connection that is supposed to be defined in the boundaries of sub-panels along knuckle lines).
- Directly after the **TYPE** statement must follow a **RADIUS** statement defining the bending tool radius, e.g. RADIUS=30.
- Then there must follow at least one thickness group. It starts with a **THICKNESS** statement giving the maximum thickness up to which (and including) the following parameters are valid, e.g. THICKNESS=10.

For a full explanation of the above statements and the methods used during the calculation of stretching compensations please refer to the Tribon M3-User's Guide, Tribon M3 Hull, Set-up and Customisation, Knuckled Pieces and Swedgings, Handling of Knuckled Panels in TRIBON, Control of Bending Characteristics.

For a customer using two different bending tools with the bending radius 30 and 40 millimetres, a control file could be defined as shown opposite:

```
TYPE=9999
RADIUS=40
THICKNESS=10
15      0.4
30      0.6
90      0.8
THICKNESS=20
15      0.5
30      0.7
90      0.9
THICKNESS=999
15      0.6
90      1.0
TYPE=33
RADIUS=30
THICKNESS=10
15      0.5
30      0.7
90      0.9
THICKNESS=20
15      0.6
30      0.8
90      1.0
THICKNESS=999
15      0.7
90      1.1
```

After creating the file and assigning it to the logical **SBH_KNUCKLE_CTRL**, reset the project and start the **Initiate Hull Standard** program.

Go to the **Knuckled Pieces** section and click on the **Knuckle ctrl, check** option.

Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.12 Folded flanges

A number of characteristics of folded flanges, e.g. the width of the area to be bent related to the nominal height, the radius of the bending tool, the shapes of the ends of the flange, may vary considerably between different yards. Therefore, a yard should be able to set up a suitable flange standard of its own. TRIBON has a facility with exactly this objective. Based on a number of basic patterns it may establish any number of flange alternatives to be used both in brackets and other plate parts.

The customer set-up of a flange standard is done via a text file in an application specific syntax based on the general TIL language for text input.

The flange definition is divided into two parts:

- The definition of the flange type.
- The definition of flange ends.

The **flange type** describes parameters common to the whole flange, e.g. width of the flange, the position of the marking line (the line indicating where the bending tool should be placed) and the bending radius. Each flange is assigned a user defined type number. Normally, a customer would have only one flange type.

The **flange end type** describes the shape of the ends of the flange. These definitions are based on predefined basic types. By setting the parameters of the basic flange ends types, each customer creates his own set of standardised flange ends.
A folded flange is not necessarily symmetric. The flange ends may be different, while - of course the total height of the flange and the position of the marking line must be independent of the flange ends. Normally, any combination of flange type and flange ends is allowed.
When generating a flange the user selects a certain flange type and also specifies the (user defined) flange end types. If not specified the user defined default types will be used.

The order of the statements in the flange control file is arbitrary. Note, however, that the default flange type and flange end type will depend on the order if not otherwise specified. A flange definition file must contain at least one flange type and one flange end type, but otherwise the customer may define as many flange types and flange end types as he wishes.

Below is an example of a flange definition file:

```
FLA_TYPE, 1
        /RADIUS             = 25
        /W_H_FACT           = 1.0
        /M_CONST            = 14;

FLA_TYPE, 2
        /DEFAULT
        /RADIUS             = 25
        /W_H_FACT           = 1.0
        /W_RAD_FACT  = 0.6
        /M_CONST            = 0;

FLA_END, 100
        /DEFAULT
        /TYPE               = 1;

FLA_END, 3
        /TYPE               = 3
        /KN_RAD_FACT = 1.2
        /KN_T_FACT          = 0.5
        /KN_CONST           = 10
        /FLA_V       = 30
        /R1                 = 20;

FLA_END, 9
        /TYPE               = 9
        /KN_RAD_FACT = 1.2
        /KN_T_FACT          = 0.5
        /KN_CONST           = 10
        /FLA_V       = 30
        /R1                 = 200
        /R2                 = 60
        /TOE_H       = 15
        /TOE_L       = 40
        /DX_P               = 100
        /DY_P               = 150
        /DIST               = 20;
```

For a complete explanation of all of the terms shown above, and an explanation of the calculations used when generating folded flanges please refer to the Tribon M3 User's Guide, Tribon M3 Hull, Set-up and Customisation, Flanges, Folded Flanges in Tribon.

After creating/updating the file and assigning it to the logical **SBH_FLANGE_CTRL**, reset the project and start the **Initiate Hull Standard** program.

Go to the **Profiles and Flanges** section and click on the **Folded flange, create** option.

Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create an object called **__SBH_FLANGE_CTRL__** in the **SB_OGDB**.

If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.13 Material Qualities

3.13.1 Defining Qualities

TRIBON Hull allows the user to specify the material quality (grade) of all parts when they are modelled. The assigned material qualities must all have been registered beforehand via a user defined text file. The total file specification of which should be assigned to the TRIBON environment variable **SBH_QUALITY_LIST**.

The layout of the file is described below. The format is free, but it is recommended to let each quality be defined in a line of its own. The line width is restricted to 80 characters. Any number of qualities may be defined in the one file.

```
'<quality1>'   <int_code1>   <density1>   ['<paint1>';]
'<quality2>'   <int_code2>   <density2>   ['<paint2>';]
'<quality3>'   <int_code3>   <density3>   ['<paint3>';]
```

<quality> The identification of certain quality. This is the 'name' by which a certain quality is picked. Its length is restricted to 24 characters.

<int_code> An integer number is used internally in TRIBON to represent a certain quality string. It must be unique within the file and **must not be changed during a project.**

<density> The density of this material. Should be expressed in kg/mm3.

<paint> This is an optimal parameter that, when given, should be followed by a semicolon. It may be used to specify special painting to be associated with certain material quality. The length is restricted to 26 characters.

- The densities can be changed at any moment. The new value will come into operation the next time the material is accessed.
- **<paint i>** can be changed at any moment. The new string will come into operation the next time the paint information is accessed. New qualities can be added at any moment.
- For a material quality with the default density the density need not be given explicitly but may be replaced by an asterisk (*).
- The quality with **<int_code>=0** will be output as the default quality, i.e. the quality you get if not explicitly defined.
- The default density is 0.00000786 kg/mm^3. Defining the quality **DEFAULT** in the quality file can change the default density. The definition should be made in the beginning of the file and should occur only once.
- There exists no default paint, but the **DEFAULT** statement can be used to define it.

An example of a quality set-up file is shown below:

```
DEFAULT -9999   8.00E-6  Default paint;
A        0   *      A-paint;
A27     1027   *      A27-paint;
A32     1032   *
A36     1036   *
D      4000   7.86E-6
D27     4027   7.86E-6
D32     4032   7.86E-6
D36     4036   7.86E-6
A27Z    1279   *
STE305   9305   *
ALU     30000   2.70E-6
```

After creating the file and assigning it to the logical **SBH_QUALITY_LIST**, reset the project and start the **Initiate Hull Standard** program.

Go to the **Material Qualities** section and click on the **Qualities, check** option.

Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.13.2 Quality exchange

The TRIBON Nesting system will allow the nesting of certain material qualities on different material quality raw plates. However before this can happen the system has to be informed which material qualities are an acceptable alternative to the original material quality.

This information is provided in a text file assigned to the logical variable **SBH_QUALITY_EXCH**. In this file each material quality should be listed along with a list of raw plate material qualities the part can be nested on.

For each quality, the acceptable exchange qualities are given in a list separated by a comma (,) and terminated by a semicolon (;). The order within the list is arbitrary.

An extract from a quality exchange file is shown below:

```
A,    A,D,E,A27,A32,A36,D27,D32,D36,E27,E32,E36,
      A27Z,A32Z,A36Z,D27Z,D32Z,D36Z,E27Z,E32Z,E36Z,
      RST37-2,ST37-3,ST44-2,ST44-3,ST52-3,STE255,STE285,STE315,STE355;
D,    D,E,D27,D32,D36,E27,E32,E36,D27Z,D32Z,D36Z,E27Z,E32Z,E36Z,
      ST37-3,ST44-3,ST52-3,STE255,STE285,STE315,STE355;
E,    E,E27,E32,E36,E27Z,E32Z,E36Z,STE255,STE285,STE315,STE355;
A27,  A27,A32,A36,D27,D32,D36,E27,E32,E36,
      A27Z,A32Z,A36Z,D27Z,D32Z,D36Z,E27Z,E32Z,E36Z,
      ST44-2,ST44-3,ST52-3,STE285,STE315,STE355;
A32,  A32,A36,D32,D36,E32,E36,A32Z,A36Z,D32Z,D36Z,E32Z,E36Z,
      ST52-3,STE315,STE355;
A36,  A36,D36,E36,,A36Z,D36Z,E36Z,ST52-3,STE355;
D27,  D27,D32,D36,E27,E32,E36,D27Z,D32Z,D36Z,E27Z,E32Z,E36Z,
      ST44-3,ST52-3,STE285,STE315,STE355;
D32,  D32,D36,E32,E36,D32Z,D36Z,E32Z,E36Z,ST52-3,STE315,STE355;
D36,  D36,E36,D36Z,E36Z,ST52-3,STE355;
E27,  E27,E32,E36,E27Z,E32Z,E36Z,STE285,STE315,STE355;
E32,  E32,E36,E32Z,E36Z,STE315,STE355;
E36,  E36,E36Z,STE355;
A27Z, A27Z,A32Z,A36Z,D27Z,D32Z,D36Z,E27Z,E32Z,E36Z;
A32Z, A32Z,A36Z,D32Z,D36Z,E32Z,E36Z;
A36Z, A36Z,D36Z,E36Z;
D27Z, D27Z,D32Z,D36Z,E27Z,E32Z,E36Z;
D32Z, D32Z,D36Z,E32Z,E36Z;
D36Z, D36Z,E36Z;
E27Z, E27Z,E32Z,E36Z;
E32Z, E32Z,E36Z;
```

After creating the file and assigning it to the logical **SBH_QUALITY_EXCH**, reset the project and start the **Initiate Hull Standard** program. Go to the **Material Qualities** section and click on the **Quality exchange, check** option. Use the **Check** button and the system will check your file for syntax errors. If any errors are found use the **Edit** button and make the necessary changes before checking the file again.

3.14 Customising Dialogues in Tribon Hull

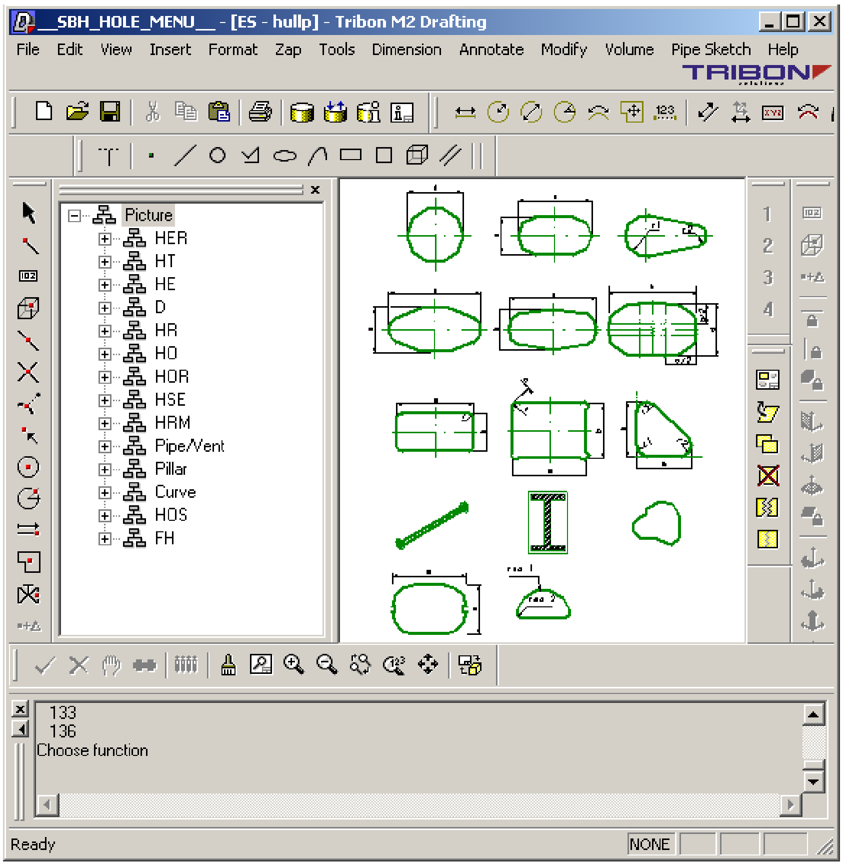
Graphical menus used in dialogues (e.g. hole type selection) are controlled and customised by updating specific drawings and certain picture object attributes connected to these drawings. There is one specific drawing with a fixed name for each dialogue and they are all stored in the databank SB_SETTINGS_DB that is accessible from the drafting application.
The template project delivered with Tribon contains examples of these drawings, that are used for such dialogues.

Below is a list of the drawings and the corresponding dialogues that are supported:

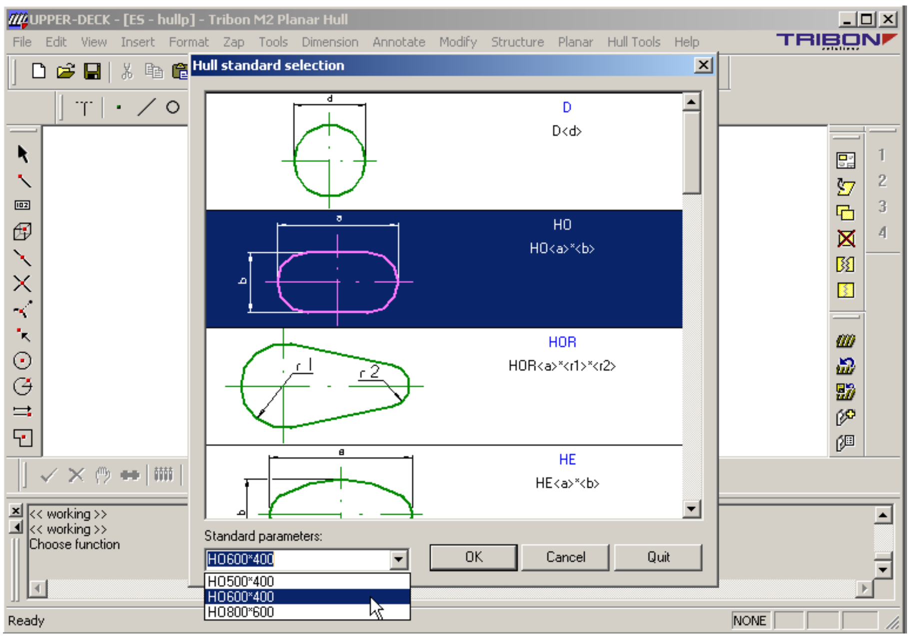
__SBH_HOLE_MENU__	When creating holes in planar panels When creating holes in planar profiles
__SBH_EDGE_NOTCH_MENU__	When creating edge notches along limits of planar panels. When creating notches in planar profiles When creating notches in shell profiles
__SBH_CORNER_NOTCH_MENU__	When creating corner notches in planar panels
__SBH_BKT_SYNT_MENU__	When selecting bracket syntax
__SBH_BKT_MENU__	When selecting a bracket

3.14.1.1 Example of hole menu customisation

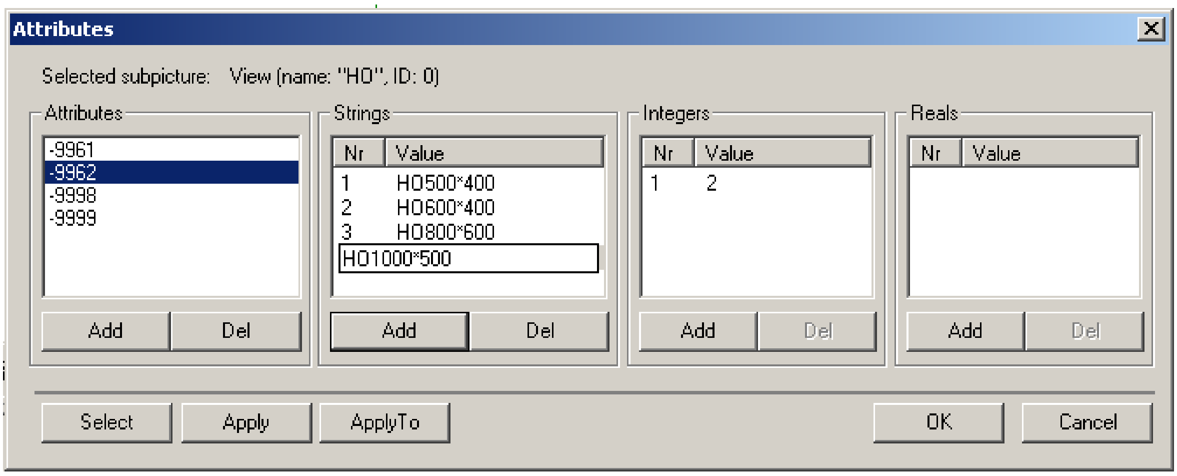
An example of the drawing used to create the hole selection menu is shown below:



The hole selection menu created from the previous drawing is shown below:



To add an instance of a hole, open the Toolkit drawing __SBH_HOLE_MENU__.
Select Tools>Inspect Drawing>10 Attributes
Select the hole geometry you wish to add an instance too. Select sub-picture level 1
Select Attribute -9962
Select the Add button in the Strings column. Key in the size of the hole (the system will automatically add a number).
Select OK. The next time you access the hole menu (shown above) the new instance will be added to the drop down selection box. (For the example shown below HO1000*500 would be added to the list).



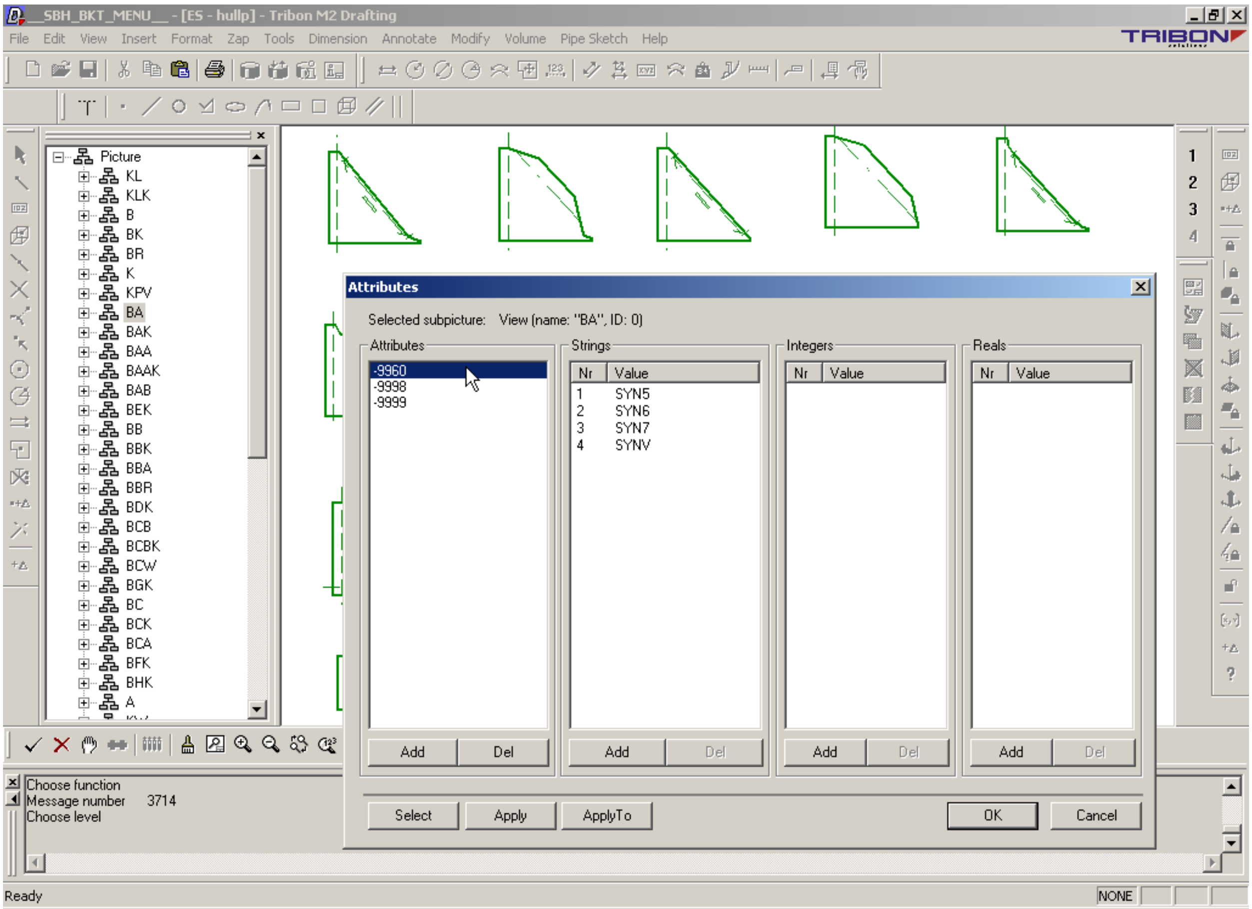
3.14.1.2 Example of bracket menu customisation

Open the Toolkit drawing __SBH_BKT_MENU__. (Display the treeview)
The bracket geometry is created on views (sub-pictures) which should named to match the bracket names defined in the bracket control file (see Section 3.6 Brackets)

If the brackets have been named correctly, when the function Hull Standards>Other Standard Set-up>Bracket std,Create is invoked and the bracket control file is selected and then Create Object is selected, if no errors are found, then the attribute –9960 will be added to the drawing __SBH_BKT_MENU__.
If you wish to change the names of brackets it must be done, in both, the bracket control file, and also in the toolkit drawing __SBH_BKT_MENU__, to change the name in the drawing select Tools>Subpicture>Rename, select view level 1, and key in the new name for the view (which should match the bracket name).

An example of the drawing __SBH_BKT_MENU__ is shown below: Attribute –9960 is highlighted and the values assigned are listed in the Strings column

To view the attributes select Tools>Inspect Drawing>10 Attributes
Select the bracket geometry which contains the attributes you wish to view. Select sub-picture level 1. Select Attribute -9960



3.15 Creating parent plates

Whilst using the Plate Nesting application, if a new nest is initiated and stored before any plate parts are added to it, it will be stored as a parent plate. This interactive method is fine for defining one or two parent plates, but if a large number of parent plates are to be defined it is too time consuming to do it this way.

Therefore it is possible to create a text file which contains the description of multiple parent plates. This file can then be processed and all the parent plates created under one function.

The following information about each plate must be given in the file:

- plate name (max 24 characters)
- length (real)
- width (real)
- thickness (real)
- quality (max 25 characters)
- density (real)
- charge (max 75 characters) may be left blank e.g. ‘ ‘
- preparation (max 25 characters) may be left blank e.g. ‘ ‘
- purchase (max 25 characters) may be left blank e.g. ‘ ‘

An example of this file can be seen below:

```
'010001A'  6.00E+03  2.00E+03  1.00E+01 'A   '  7.84000000E-06 'TEST' '' ''
'010002A'  8.00E+03  2.00E+03  1.00E+01 'A   '  7.84000000E-06 'TEST' '' ''
'010003A'  1.00E+04  2.00E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
'010004A'  1.20E+04  2.00E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
'010005A'  6.00E+03  2.50E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
'010006A'  8.00E+03  2.50E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
'010007A'  1.00E+04  2.50E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
'010008A'  1.20E+04  2.50E+03  1.00E+01 'A   '  7.84000000E-06 '' '' ''
```

Note: It may assist the handling of plates if the name includes the thickness and quality, in the example shown above the plates are 10 thk, quality A.

To process a previously defined file:

Start **Hull Nesting** & open an existing nest or start a new nest, then select **Nesting → Nesting Tools → Nest PPI → Create Standard Plate**.

The system will display a list of available files in the **SB_SHIPDATA** directory of the current project, browse to display the file you have created.Select the file and click the **Open** button. The system will now create all the parent plates contained inside the data file and write them to the nesting standards databank.

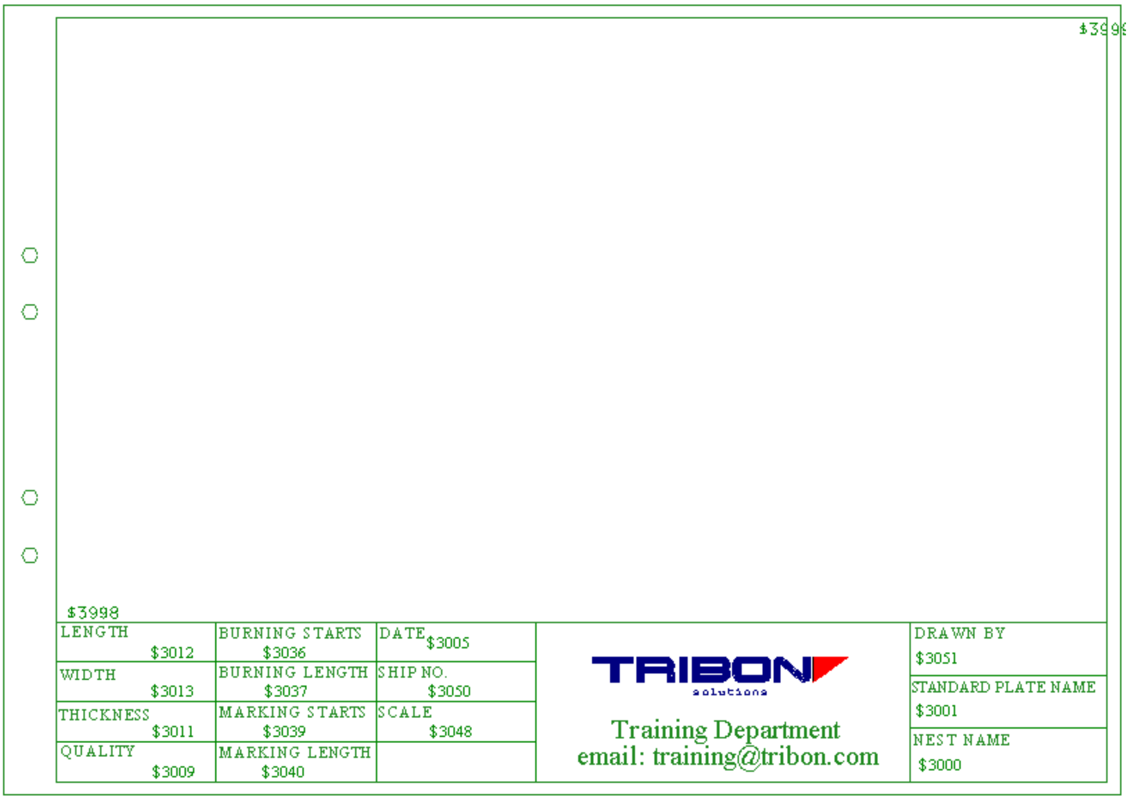
3.16 Nesting

3.16.1 Burning sketches

The available drawing forms for burning sketches are stored in the databank associated with the variable **SBD_STD**. Multiple drawing form variations can exist in this database but only one can be associated with each nest job.

The drawing form is constructed with a mixture of traditional drafting functions and the use of some simple rules.

Please consider the example below:



The lines and text are created using traditional drafting functions. The Tribon logo is placed using input sub-picture. All of the dollar values will result in values being automatically added for the current nest when the drawing form is added. Note values **$3998 & $3999** must be defined for the plate sketch to be placed in the drawing form for 2 axis burning machines. **$3998** defines the lower left corner, **$3999** defines the upper right corner. For 3 axis burning machines these values are replaced with **$9998 & $9999** respectively.

For a full list of all available dollar values please refer to the Tribon M3 User's Guide, Tribon M3 Hull, Manufacturing, Plate Nesting, Tribon Hull Plate Nesting, Nesting System-Application Functions, Functions Overview, Shop Drawing Functions.

3.16.2 Defining a new nesting drawing form

Create the geometry and text as desired using traditional drafting functions.

Start the function **Tools → Drawing Form → Save**.

The system will prompt for a name for the drawing form. Key in the desired name and click the **OK** button.

The system will store the drawing form in the **SBD_STD** and clear it from the screen.

3.16.3 Modifying an existing nesting drawing form

To modify an existing drawing form start the command

Tools → Drawing Form → Open

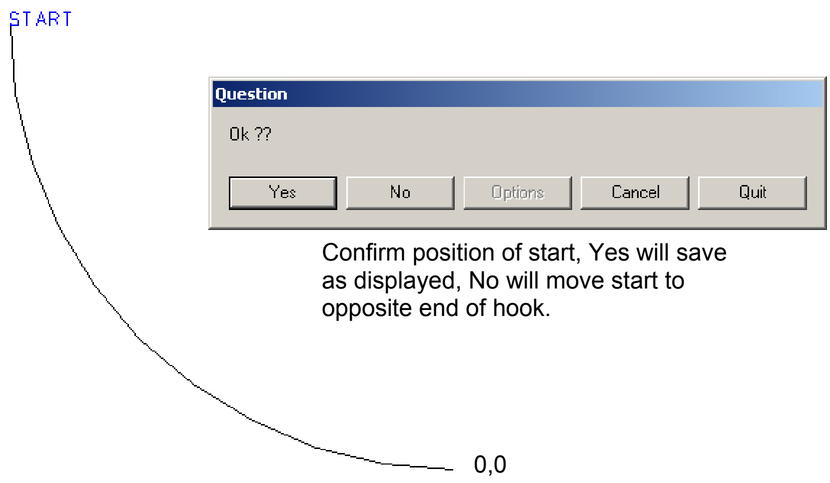
The system will display a list of existing drawing forms. If the list appears empty or contains the wrong objects clear the **Object name** field and click the **List** button. Select the desired drawing form and click the **Open** button.
Make the required changes and then use the **Tools → Drawing Form → Save**. function.
The system will proceed through the same prompts as shown when creating a new drawing form.

3.16.4 Deleting a drawing form

To delete a drawing form from the SBD_STD database, the **DBTools** application should be used.

3.16.5 Defining a new hook

- Go to drawing mode by selecting **Nesting > Nesting job > Return to Drawing**
- Start a new drawing by **File > New**
- Give the new drawing the name required for the new hook (max 5 numeric characters) e.g. 12345
- Draw the geometry of the hook using the drafting tools. Note:The end of the hook should be at the origin (0,0) with the positive u-axis as the continued tool path and the scrap side above the u-axis.
- When ready, select the function **Nesting > Nesting tools > Hooks > Save**



The hook will be saved in the object **D003HOOKS** stored in the nesting standards databank **SB_NSTD** and the drawing will be saved in the drawing databank **SB_PDB** and will be removed from the screen. The message line will confirm whether the hook has been saved or not.

3.16.6 Modifying an existing hook

- Use the function **Nesting > Nesting tools > Hooks > Open**.
- Select the hook to be modified
- Make the required changes to the hook geometry.
- Select **Nesting > Nesting tools > Hooks > Save**

The starting point of the hook will be highlighted for verification. Confirm position of start, Yes will save as displayed, No will move start to opposite end of hook.

The hook will be saved and the drawing will be removed from the display.

ⓘ *Hooks are described in their own co-ordinate system. Ending at the origin (start point) with the positive U-axis as the continued tool path and the scrap side above the U-axis.*

3.17 Automatic position numbers

The main purpose of the Automatic Position Number utility is to automatically assign position (piece) numbers to plane panel parts. In doing so the facility usually checks parts for equality and assigns the same position number to identical (within certain limits) parts. All parts that are treated are sorted according to user defined criteria before the numbering process. The position numbers themselves are just running numbers taken from a user-defined interval chosen in increasing order. This chapter deals with the setting up of the position number criteria, not the actual assignment of the position numbers themselves.

The information controlling the position numbers is stored in an object in the **SB_OGDB.** The name of this object should be allocated to the variable **SB_POSNO_SETS** in the project definition file. This variable must be assigned before creating the object as its value is used to name the object on the **SB_OGDB**.
After assigning the logical mentioned above it is necessary to create a text file to be used as input to create the position number object. An example of such a file is shown below:

```
NEW;
SYM_EQUAL;
SORT/AFT_FOR/TOP_BOT;
BUNDLE /STI/FLA/PIL/BPR;
BUNDLE /BRA/CLI;
CASE,'1' /GEO/START=1/MAX=500/PLA
         /GEO/START=501/MAX=800/STI/FLA/PIL/BPR
         /GEO/START=801/MAX=999/BRA/MAX_BRA_AREA='50';
LIST;
```

NEW: The object connected to the environment variable **SB_POSNO_SETS** will be replaced with a new object defined with the TIL-format input file.

SYM_EQUAL: It is possible to specify whether or not parts belonging to symmetrical panels should be equipped with same or individual position numbers in port and starboard occurrences. True if given.

SORT: By selecting a path of sorting attributes it is possible to define in which order the various parts should be sorted within the ship

BUNDLE: By default, all part types are treated separately, meaning that e.g. plates are numbered continuously, not interrupted by e.g. stiffeners or clips. This also means that flanges, stiffeners, pillars and bracket stiffeners are treated separately. There is however possibilities to specify that one wishes to treat two or several part types as a single type. This is achieved by part bundling. It is possible to specify a number of bundled groups, each containing an arbitrary number of part types. The meaning of these groups is that the automatic setting will treat all part types within each of the groups as one type, thereby allowing these to be mixed within the current number series (this is normally the case with stiffeners, flanges and pillars, i.e. "profiles").

CASE <case name>: It is possible to define one or several cases that can be referenced later on. It is thought that a certain case should contain:

- a user defined name that can be used when referencing it
- an arbitrary number of user defined number series

The name of a case is totally user defined, used as a tag for later references. The idea behind a number series is that it should contain:

- a certain list of part types (e.g. plates, flanges etc.)
- a starting number
- an optional maximum number
- criteria's to sort parts according to

LIST: The current continence of the object connected to the environment variable **SB_POSNO_SETS** will be listed.

DELCASE: The case with the given name will be deleted.

For a full explanation of all the possibilities available in each statement please refer to the TRIBON M3 User's Guide – Hull Manufacturing – Automatic Position Number Setting.

After creating/updating the file start the **Initiate Hull Standard** program. Go to the **Naming** section and click on the **Autopos set-up** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create an object named after the setting of **SB_POSNO_SETS** in the **SB_OGDB**.

If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.18 Part name control

Parts (e.g. plates and stiffeners) are in the production stage identified by names, which are often painted or otherwise fixed onto the parts. The names may also be issued in drawings and production lists of different kinds. These production-oriented names are in TRIBON referred to as **part names**. The part names are not equal to and should not be confused with the names of parts used within the TRIBON data banks ("TRIBON names"). There are several reasons for this. One is that identical parts must have unique names because they have individual occurrences and positions and e.g. contribute in an individual way to a centre of gravity calculation. On the other hand it may be advantageous to let equal parts have the same part names because that may reduce the handling/sorting in the workshop.

A part name normally consists of a number of **constituents** that are combined to build up the name, separated by **delimiters**. Examples of constituents are the project name, names of assemblies, position (piece) numbers, etc. Delimiters can be any fixed strings but often consist of a hyphen (-) or a slash (/).

However, it may not always be necessary to present the part name in its full (maximum) size, but only a part of it. E.g. all parts in a drawing may belong to the same assembly and therefore it is enough to present the position (piece) number to uniquely identify the part within the drawing. Thus there is a need for a customer to set up rules for **different levels** of the part name for one and the same part.

Depending on the environment or context alternative layouts can be chosen. There may be different rules for the names of e.g. normal plates, brackets plates and stiffeners. Therefore there is a need that part names should be controlled **individually** for **different types** of parts.

The length of a part name is restricted to 48 characters. However, certain applications do not accept a part name of that length. It is recommended to restrict the part name length to 24 characters, if possible.

The customer specification of the name rules is set up in a text file and this should be assigned to the logical variable **TB_PARTNAME_CTRL**. This text file will then be used to create an object named **__TB_PARTNAME_CTRL__** on the **SB_OGDB**. An example of such a file is shown below:

```
LEVEL,FULL;
LEVEL,SHORT;
! "FULL" part names
HPPL/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HSPL/ LEVEL=FULL/ SBLOCK/ DEL='-‘/ POS;
HBPL/ LEVEL=FULL/ SBLOCK/ DEL='-B'/ POS;
HCPL/ LEVEL=FULL/ SBLOCK/ DEL=’-‘/ POS;
HPPR/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HSPR/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HBPR/ LEVEL=FULL/ SBLOCK/ DEL='-B'/ BPOS/ LPOS;
! "SHORT" part names
HPPL/ LEVEL=SHORT/ POS;
HSPL/ LEVEL=SHORT/ POS;
HBPL/ LEVEL=SHORT/ DEL='B'/ POS;
HCPL/ LEVEL=SHORT/ POS;
HPPR/ LEVEL=SHORT/ POS;
HSPR/ LEVEL=SHORT/ POS;
HBPR/ LEVEL=SHORT/ DEL=’B’/ BPOS/ LPOS;
```

Suppose the above rules are used and applied to a block named AA123-4. Also assume all position numbers equal 1, then the resulting part names would be as follows:

	Full part names	**Short part names**
Plane plate	**123-4-1**	**1**
Shell plate	**123-4-1**	**1**
Bracket plate	**123-4-B1**	**B1**
Clip plate	**123-4-1**	**1**
Plane panel profile	**123-4-**	**1**
Shell profile	**123-4-1**	**1**
Bracket profile	**123-4-B1A**	**B1A**

After creating/updating the file and assigning it to the logical **TB_PARTNAME_CTRL** reset the project and start the **Initiate Hull Standard** program. Go to the **Naming** section and click on the **Part name control** option. Use the **Browse** button and locate the file. When the file is located use the **Create Object** button. The system will attempt to create the object from the file selected. If the system returns any errors use the **Edit TIL File** button to modify the file as required before using the **Create Object** button again.

3.18.1 Part name level

Any number of partname control rules (=levels) may be defined as described above. But by default only two levels may simultaneously be used in the different modules of Tribon (as described in relation with TB_PARTNAME_FULL and TB_PARTNAME_SHORT above).
This paragraph describes a feature by which a certain name rule (="level") may be defined to be used within a certain hull module. (This feature has currently the restriction that only one level can be defined for a certain module, i.e. it is not possible to control different rules in different functions of a module).
This name rule section is facilitated by a simple text file whose total file specification should be assigned to the Tribon environment variable TB_PARTNAME_LEVEL. The layout of this file is very simple:

The file consists of pairs of module name and level name, one per line, separated by a comma (,) or blank. See below.

```
PROFNEST, RULEB
ASS_PARTNAME, RULEA
```

This means that level RULEA will be used when creating partnames in Assembly and Weld Planning, RULEB in profile nesting.

The following module names are currently valid (default level is FULL unless otherwise specified):

ASS_PARTNAME	Assembly Planning and Weld Planning
ASSADG_PARTNAME	Assembly Planning – Automatic Drawing
DATA_EXTR	Hull Data extraction (extraction of part identification).
DM_TREE	Design Manager based applications.
DRAW_MODEL	Draw model geometry (in symbolic views of plates, stiffeners, flanges, pillars and brackets).
LABEL_PLCM	For labelling of port names in PLCM
LABEL_NEST2AX	For labelling of port names in nesting
MARKING_NEST2AX	For marking of names (“position numbers”) along marking lines in nesting. Default: SHORT
MARKING_PLCM	For marking of names (“position numbers”) along marking lines in PLCM. Default: SHORT
MODEL|_INFO	Hull model info (when asking for information on any of boundaries, plates stiffeners, flanges, pillars and brackets).
PART_NEST2AX	For part names of plates in nesting sketches.
PART_PLCM	For part names of plates in PLCM sketches.
PROFILE_SKETCH	Profile names in sketches and listings.
PROFNEST	Automatic nesting of profiles (part names in list files).
TBRPLATEINT	Plate interface (part names).
TBRPROF_CUTINT	Profile interface lists and sketches (for marking on profiles).

The partname control file described in 3.1.8 would then be revised to include the new rule definitions as shown below:

```
LEVEL,FULL;
LEVEL,SHORT;
LEVEL,RULEA;
LEVEL,RULEB;
!
! "FULL" part names
!
HPPL/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HSPL/ LEVEL=FULL/ SBLOCK/ DEL='-‘/ POS;
HBPL/ LEVEL=FULL/ SBLOCK/ DEL='-B'/ POS;
HCPL/ LEVEL=FULL/ SBLOCK/ DEL=’-‘/ POS;
HPPR/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HSPR/ LEVEL=FULL/ SBLOCK/ DEL='-'/ POS;
HBPR/ LEVEL=FULL/ SBLOCK/ DEL='-B'/ BPOS/ LPOS;
!
! "SHORT" part names
!
HPPL/ LEVEL=SHORT/ POS;
HSPL/ LEVEL=SHORT/ POS;
HBPL/ LEVEL=SHORT/ DEL='B'/ POS;
HCPL/ LEVEL=SHORT/ POS;
HPPR/ LEVEL=SHORT/ POS;
HSPR/ LEVEL=SHORT/ POS;
HBPR/ LEVEL=SHORT/ DEL=’B’/ BPOS/ LPOS;
!
```

```
! "RULEA" part names
!
HPPL/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HSPL/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HBPL/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HPPR/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HSPR/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HBPR/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
HCPL/ LEVEL=RULEA/  ASS=1/ DEL='-'/ POS /SYM;
!
!
! "RULEB" part names
!
HPPL/ LEVEL=RULEB/ POS /SYM;
HSPL/ LEVEL=RULEB/ POS /SYM;
HBPL/ LEVEL=RULEB/ POS /SYM;
HPPR/ LEVEL=RULEB/ POS /SYM;
HSPR/ LEVEL=RULEB/ POS /SYM;
HBPR/ LEVEL=RULEB/ POS /SYM;
HCPL/ LEVEL=RULEB/ POS /SYM;
!
```

For a full explanation and description of all available options regarding the setting up of part names, please refer to the TRIBON M3 User's Guide – TRIBON M3 Set-up and Customising – General – Customer Control of Part Names.