

Some handy attributes

1 Adding parameters to Transverse planes and Horizontal planes

The user must be able to define a table containing a number of transverse reference planes. In the [Domain Expert tutorial](#), the purpose of entity Transverse planes is described in detail.

For each plane the user has to define a position and a name.

- Add the following parameters in the **Knowledge Browser**:

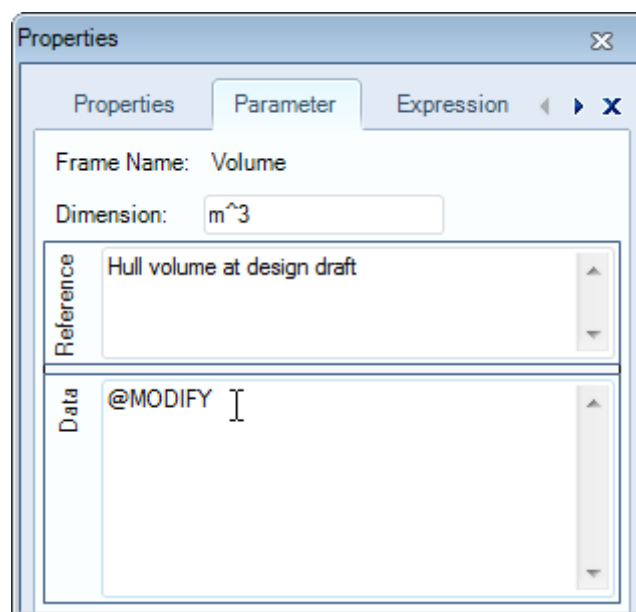
Parameter name	Dimension	Determined by	Reference	In Class
Nr	[#]	VR: User only	Number of instances	General
Name\$	[Str]	VR: User only	Name of object	General
CaseID	[-]	USR: User or system/equation	Case index	General
Frame_Nr	[-]	VR: User only	Frame number	General
Frame_spacing	[mm]	VR: User only	Frame spacing	Dimensions
X	[m]	VR: User only	X position, in longitudinal direction	Dimensions
Z	[m]	VR: User only	Z position, in vertical direction	Dimensions

- Drag and drop parameters Nr, Names\$ and CaseID in both Horizontal planes and Transverse planes.
- Drag and drop parameters Frame_Nr, Frame_spacing and X in Transverse planes.
- Drag and drop parameter Z in Horizontal planes.

2 Enable users to modify calculated values

In the [Domain Expert version](#) of the tutorial, it was shown that sometimes a user can modify calculated values after they are determined by the system. You, as a Knowledge Engineer, can enable this functionality by adding a **@MODIFY** attribute to the data slot of the parameter.

- Click on the parameter Volume in the Knowledge Browser, select the *Parameter* tab of the **Properties** window and enter **@MODIFY** in the *Data* field.

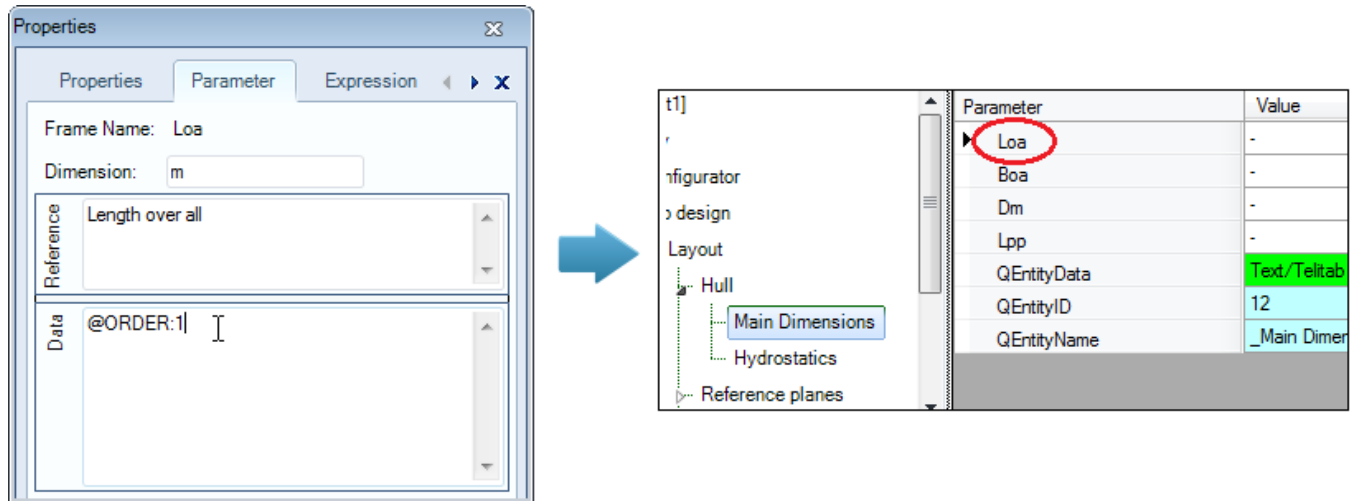


3 Modify parameter display order

It can be desirable to influence the parameter display order in the **Workbase** list. Normally, Quaestor puts parameters in alphabetical order. An **@ORDER** attribute on a parameter defines the relative position of the parameter in the **Workbase** list. Define this position by: **@ORDER:RelativePosition**, in which **RelativePosition** is an integer. The **@ORDER** values need not be consecutive. Parameters are sorted on **RelativePosition** from low to high. Values with no **@ORDER** attribute are placed below parameters with **@ORDER** attributes.

Here, as an example, you will place parameter **Loa** at the top of the list. It is up to you to order the other parameters.

- Click on the parameter **Loa** in the **Knowledge Browser**, select the **Parameter** tab of the **Properties** window and enter **@ORDER:1** in the **Data** field.

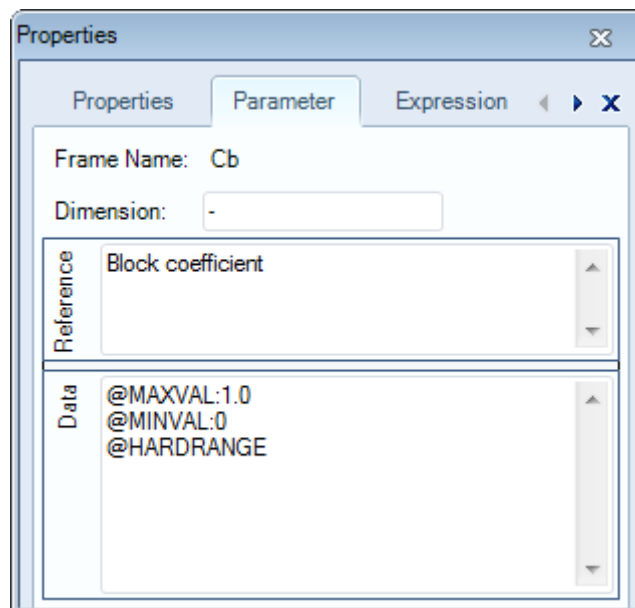


4 Define minimum and maximum values for parameters

Another interesting feature to assist the user, is to provide feedback about minimum and maximum input values. Moreover, you can make these boundaries into hard ranges in order to prevent the user from providing values that might cause faulty results.

As an example, you can define a minimum and a maximum for the parameter **Cb**. This value must lie between 0 – 1.0. The system should issue a warning if the computed or input value is not within this range. This is done by means of the attributes **@MAXVAL** and **@MINVAL**. If you want the system to force a value inside this range, add **@HARDRANGE** as well.

- Add the following attributes to parameter **Cb**: **@MAXVAL:1.0**, **@MINVAL:0**, **@HARDRANGE**



5 Have a parameter determine the number of cases in an entity

Parameter Nr will be used to indicate the number of cases (columns) in an entity.

- Provide parameter Nr with the special attribute **@NRINST** (Data field of *Parameter* tab in **Properties** window).

The **@NRINST** attribute tells Quaestor that the value given/calculated for this parameter indicates the number of instances/cases in an object (in this case an entity).

It is useful to also provide **Nr** with the attribute **@INTEGER**. The **@INTEGER** attribute limits input or computed values of the parameter to integer values. If a non-integer value is either computed or provided, the system issues a warning and prompts for other input or for new input of other parameter values that lead to this result. In addition, for this integer value you should change the number of decimal places to 0.

- Perform the necessary modifications.

6 Make parameters multicase

Normally, the value of any single value parameter is displayed in the **list view**.

The **@MULTVAL** attribute forces a single value parameter to present itself in table form. The **@MULTVAL** attribute is used in parameters if you wish to obtain a table which includes all values of these parameters whether they are single values or not. See also the [Domain Expert tutorial](#).

The following parameters need to be shown in the table: Name\$, CaseID, Frame_Nr and X.

- Set a **@MULTVAL** attribute on these parameters.

By doing this, you can see these parameters being moved from **list view** to **table view** (you might have to refresh Quaestor with **Ctrl+U** to redraw the **Workbase**).

The screenshot shows the Quaestor Workbase interface. On the left is a tree view of the project structure. On the right is a table view showing parameters and their values.

Parameter	Value	Dimension
Frame_spacing	-	mm
Nr	-	#
QEntityData	Text/Telitab	Str
QEntityID	14	m
QEntityName	_Transverse planes	Str

	#1
CaseID [-]	-
Frame_Nr [-]	-
Name\$ [Str]	-
X [m]	-

7 Provide default values

- Create the following relation for parameter X in entity Transverse planes in the **Workbase**:

$$X = \text{Frame_nr} * (\text{Frame_spacing}/1000)$$

For each case, representing a transverse reference plane, this relation will be calculated. Parameter Frame_spacing does not have a **@MULTVAL** attribute, thus the value provided for Frame_spacing will be constant for each case.

It is possible to provide a default value for a parameter, that can be overruled by the user.

- Enter 700 in the Value cell of Frame_spacing.

8 Use of **ORCA()** to generate case numbers

- Create the following relation for parameter CaseID in entity Transverse planes in the **Workbase**:

CaseID = ORCA(1)

The function **ORCA(1)** returns the current case number (during execution). Later on, this calculated value is used to refer to one of the transverse reference planes.

Incidentally, you could also have made this relation in the **Knowledge Browser** and connected it to entity's parameter.

9 Parameters to show extra information

Every entity is automatically created with three parameters, QEntityData, QEntityID and QEntityName. These are hidden from the user output by default. The first one can show behind-the-scenes computed values. In addition, parameters QEntityDoc and QEntityRef can also show extra information.

- Include QEntityDoc and QEntityRef in the Hydrostatics and Transverse planes entities by drag/drop from the Knowledge Browser.
- Set the attribute **@SHOW** on QEntityData, QEntityID and QEntityName.

You can add a picture in QEntityDoc, which explains definitions of reference planes and the coordinate system of the vessel to the user in the **Explanation window**.

- *Right-click* on QEntityDoc in entity Transverse planes and select *Taxonomy>Include Binary Data* or press *Ctrl+B*. Select the file that you want to include (e.g. [reference_planes.bmp](#)).

Comparable to assigning a reference to a parameter, you can assign a reference to an entity. As the value QEntityRef you can enter an unlimited entity reference text, which will be shown in the html **Explanation window**. For example; "Define number of transverse reference planes"

10 Make a parameter type local to an entity

You can use the **@WBNAME** parameter attribute to define a display name that differs from the name of the parameter itself. For example, you want to define a display name **Number of transverse reference planes** for parameter Nr. Because you want this name to be unique for entity Transverse planes you cannot simply add the attribute to the data slot of the parameter!

- *Right-click* on parameter Nr in entity Transverse planes and select *Taxonomy>Instantiate "Nr"* (or press *Ctrl+E*).

Now, the background of this parameter's **Properties** window changes from white to light yellow. This means that you are now able to set properties and provide a reference text and attributes for the parameter which differ from the global reference text and attributes for this parameter.

You can restore the parameter to the global settings by again pressing *Ctrl+E* or selecting *Taxonomy>Set global "Nr"* from the context menu.

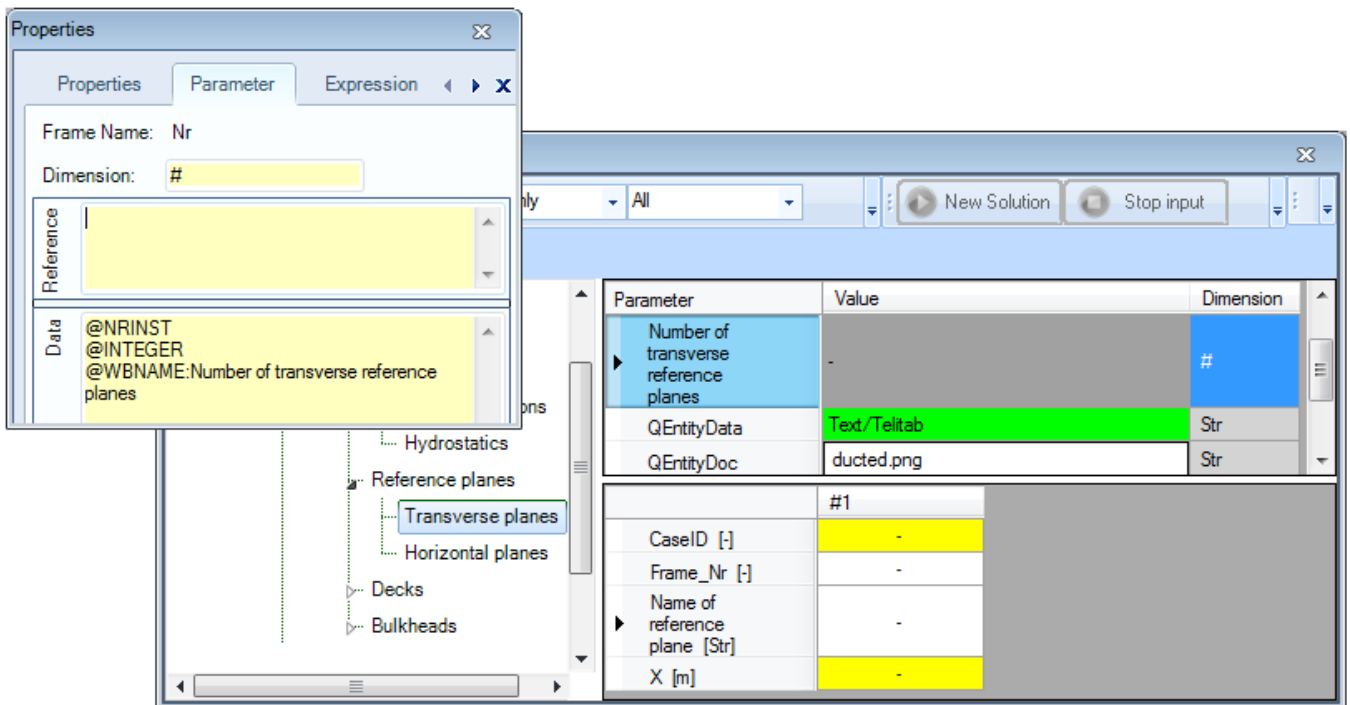
After making parameter Nr in entity Transverse planes local in this way:

- Set the attribute **@WBNAME: Number of transverse reference planes**
- Change the reference text to **Number of transverse reference planes**.
- Do the same for parameter Name\$ (make it local, **@WBNAME: Name of transverse reference plane**, reference: Name of transverse reference plane)



Remember that you can define a display sequence of the parameters in a specific entity by setting **@ORDER** attributes for localized parameters.

Finally the entity Transverse planes should look like this:



11 Completing Horizontal planes

Here is a summary of what to do to complete `Horizontal planes` in the same way.

The main goal here is that a user must be able to define a table with a number of horizontal reference planes. For each plane the user must define a position and a name.

- Provide parameters `Z`, `CaseID` and `Name$` with the attribute `@MULTVAL`.
- Include the following parameters in entity `Horizontal planes`: `QEntityDoc` and `QEntityRef`.
- Localize the parameters `Nr` and `Name$` in entity `Horizontal planes` and provide for both parameters a `@WBNAME` attribute to define a display name.
- Create the following relation `CaseID = ORCA(1)`.

This is the same relation as in `VerticalPlanes`' `CaseID`. You could also have used a global (defined in the Knowledge Browser) relation for both. Now both types of planes have each their own relation. In this example, there is not much difference. However, in more advanced usages, there are cases in which there is a significant difference.

- To show computed values, add the `@SHOW` attribute to behind `QEntityData`.
- Add a picture (e.g. `reference_planes.bmp`) as Binary to `QEntityDoc`.
- Assign the following text to `QEntityRef`: Define number of horizontal reference planes.

Now your developed entity `Horizontal planes` should look something like this:

Workbase

Local
(Internal) Name only
All
New Solution
Stop input

Dataset[Ship_configurator]

- QTaxonomy
 - Ship configurator
 - Ship design
 - Lay out
 - Hull
 - Reference planes
 - Transverse planes
 - Horizontal planes
 - Decks
 - Bulkheads

Parameter	Value	Dimension
Number of horizontal reference planes	-	#
QEntityData	Text/Telitab	Str
QEntityDoc	reference_planes.bmp	Str
	#1	
Name of horizontal reference plane [Str]	-	
CaseID [-]	-	
Z [m]	-	

[Back to content](#) |
 [<< Previous](#) |
 [Next >>](#)