Knowledge Browser

In Quaestor, the Knowledge Browser provides access to the knowledge assembled in the knowledge base and the functionality of Quaestor. Together with the Workbase the Knowledge Browser is the most significant component of the user interface and offers all the necessary possibilities to adapt search and even combine knowledge databases. Basically, it consists of seven parts.

blocked URL

1. The **Search Field** is a dynamic and powerful way of searching the knowledgebase. While typing here, field 7 shows all frames (partially) corresponding to your search string.

2. The **Switch View Buttons** are the back and forward buttons of the Knowledge Browser, comparable to those in standard HTML-browsers. By pushing the button between the back and forward buttons ('Class List'), field 3 can be switched between a tree view mode , list view mode (default) and domain view mode (see below).

3. The **Class Tree** shows the knowledgebases currently opened, and the classes they contain. Furthermore, it provides access to the integrated support system by means of the *Quaestor node*. Show flash demo...

4. The **Frame in focus/Browse field** window shows the frame (parameter, relation, constraint,...) currently selected in the knowledge browser. Clicking once on a parameter, makes field 7 show the *Expressions in which the current parameter is present*, double clicking makes sure it shows the *relations that can be used to determine current Parameter.*
5. The **View description** describes what is shown in field 7.

6. The **View Options** pulldown menu provides extended options to the data shown in field 7. For example, one might only want to view parameters in relations, containing attributes or subgoals etc.

7. The **Parameter, Relations and Constraints** window shows the contents of the knowledgebase, depending on the view options selected in field 4 and 6. It's the central place to add and edit knowledge.

Together with the Frame Viewer and Properties, the Knowledge Browser is the most significant component of the user interface for a Knowledge Engineer and offers all the necessary possibilities to search, view and modify knowledge (with the proper user rights).

Also go to Interface KE for a general explanation of all other windows.

Graphical feedback

In the field 7 all knowledge is presented. It is presented by its name and an icon communicating about the type of frame:

- Intrinsic functions: blocked URL.
- Attributes: blocked URL.
- Dimensions: blocked URL.
- Constants: blocked URL.
- Reserved or standard parameters: blocked URL.
- Normal parameters: blocked URL or blocked URL. The red cross means that additional information is required to make the parameter properly defined.
- Top goal (after double clicking a goal parameter): blocked URL.
- Normal relations: blocked URL or blocked URL. The red icon means that the relation is either switched of or corrupt.
- Constraints: blocked URL.
- Taxonomy parameters: blocked URL.
- Taxonomy entities: blocked URL.
- Taxonomy relations: blocked URL.
- Optional library functions: blocked URL. This information is only shown in a Library node below the Quaestor node when a library is available.
- Optional library attributes: blocked URL. Again this information is only shown in a Library node below the Quaestor node when a library is available.

In this way while browsing the knowledge you can identify, for instance, whether a parameter or expression is globally defined or is connected to an Entity.

Searching

To search for knowledge simply type anything in field 1 of the knowledge base. While typing, field 7 will follow by presenting the following frame types (in the given order):

1. Taxonomy parameters: blocked URL.
2. Taxonomy entities: blocked URL.
3. Normal parameters: blocked URL or blocked URL. The red cross means that additional information is required to make the parameter properly defined.
4. Constants: blocked URL.
5. Reserved or standard parameters: blocked URL.
6. Optional library functions or attributes: blocked URL or blocked URL.
7. Intrinsic functions: blocked URL.
8. Attributes: blocked URL (note that if you start typing with an @ sign you will only get attributes).

If you start with an @ sign you will see only attributes will be shown, and starting with a Q will show all constant and reserved parameters, of course in addition to functions and parameters starting with a Q.

You see that (Taxonomy) relations and constraints will not be presented in field 7. These can be found by means of the browse functionality described hereafter. By means of browsing (selecting an item in field 7 and then selecting the browse field 4), you can find connected additional information such as:

1. Taxonomy relations: blocked URL.
2. Normal relations and Constraints: blocked URL, blocked URL or blocked URL.

In addition to searching on frame names, you can search on any text part in these frames by providing an enter after typing the search text. Type something in field 1 and then give an enter and in field 7 any parameter function or attribute containing this text will be shown.

The following flash demo gives a short demonstration, just try it for yourself: Show flash demo...

Browsing

Using field 4 you can present the type of knowledge in field 7 (parameters, constraints, relations, etc. etc.). Typical selection sequences of items in the knowledge browser will enable you to visualize the dependency between parameters, relations and constraints.

With improving the attribute integration and the introduction of the taxonomy concept. This information is added in the same sequences.

Below typical actions are described. The first four flash demo are based on a Quaestor version without the above mentioned Taxonomy and attributes improvements. We have kept these demo's because the actions shown did not really change. For the Taxonomy and attribute improvement additional flash demo's are made.

Parameter to relations:

1. Select a parameter in field 7: the class it is part of is shown in field 3 and its name is shown in field 4;
2. Click on field 4: all relations the parameter is part of are shown in field 7 (and since 2.47.2 the parameter itself is presented also);
3. Double click on field 4: all relations that can determine the parameter are now shown in field 7.

Show flash demo...

Relations to parameters:

1. Double click on a class in field 3: all relations in the class are shown;
2. Select a relation in field 7, the class it is part of is shown in field 3 and its short presentation of the relation is shown in field 4;
3. Click on field 4: all parameters (and constraints, if any) related to the relation are shown.

Show flash demo...

Relations to constraints to parameters:

1. Double click on a class in field 3: all relations in the class are shown;
2. Double click on relation in field 7 having a constraint, field 7 shows the constraints;
3. Select a constraint in field 7, the short presentation of the constraint is shown in field 4;
4. Click on field 4: all parameters and relations connected to the constraint are shown.

Show flash demo...

Constraints to relations to parameters:

1. Use field 6 to show all constraints;
2. Double click on constraint in field 7 connected to a relation, field 7 shows the relations
   (or select the constraint in field 7 and go to field 4 to see the containing parameters and connected relation in field 7);
3. Select a relation in field 7, the short presentation of the relation is shown in field 4;
4. Click on field 4: all parameters and constraints related to the relation are shown.

Show flash demo...

Frames related to a function:

1. Type the name of an intrinsic function or select a function from the overview in the Quaestor node;
2. Select the function in field 7, the short presentation of the function is shown in field 4;
3. Click on field 4: all (taxonomy) relations are shown containing the function.

Show flash demo...

Frames related to an attribute:

1. Type the name of an attribute by starting with @, or select an attribute from the overview in the Quaestor node;
2. Select the attribute in field 7, the short presentation of the function is shown in field 4;
3. Click on field 4: all (taxonomy) parameters and relations are shown containing the attribute.

Show flash demo...

The Taxonomy tree, Taxonomy relations and Taxonomy parameters:

In Taxonomy applications, the Taxonomy-Entity structure is created in the dataset of the Workbase. Furthermore, Taxonomy relations and parameters are instantiated in a specific location in the Taxonomy tree.

Therefore, browsing Taxonomy related knowledge in the Knowledge Browser will result in interaction with the QTaxonomy node in the Dataset of the Workbase (or the solution based on this Taxonomy when "Show hidden data" in the Tools>Options... window is not selected) to show where the parameter or relation is instantiated.

Browsing for taxonomy parameters and relations in the Knowledge Browser works the same as for normal parameters and relations. The flash demo below show the connection between browsing in the Knowledge Browser and the Taxonomy tree in the Workbase (with "Show hidden data" in the Tools>Options... switched on).

Furthermore, the demo show that switching on the option "Connect Workbase to Knowledge Browser" under the Appearance tab of the Tools>Options window will enable you to see the (Taxonomy) relations in the Knowledge Browser related to the selection in the QTaxonomy node.

Flash demo of a Taxonomy knowledge base with Workbase-Knowledge Browser connection: click here....

Editing

Browse either to the relation, parameter or constraint and press F2.

The parameter can be renamed or the expression editor will open with the relation or constraint.

You can also remove frames. However, **a knowledge base is always protected against this modification** as this might have influence on the projects used with the knowledge base. "Under water" the frames are represented by numbers (pointers), when you delete a frame, these pointers are renumbered. And as a result projects with this old numbering will not work properly anymore (you will be warned). Therefore, when you want to remove a frame, you have to remove the delete protection (File>Allow deleting frames).

Furthermore, you are able to group relevant parameters and relations in classes in the class tree. You can create a new class and arrange existing classes in the class tree on the left side of the Knowledge Browser and drag and drop parameters and relations in these classes.
The purpose of classes is to organise your parameters. You do this by selecting any existing class and press the right mouse button.

A window will open showing the selected class with the following syntax:

Class.Sub-Class.Sub-Class….

blocked URL

In this window the class "Top Goals/Undifined" is selected.

By adding a class after the dot in the dialog you are adding a sub-class in the selected class. When you remove all and add a name, you add a class on the highest level.
Please note that any knowledge will still be available for the classes you have deleted. The information will be moved to the "Top Goals/Undefined" class.

You can add parameters to the classes by means of drag and drop actions: Select a parameter, hold your left mouse button, move your mouse (now appearing with an hand) to the class and release your left mouse button to drop it in the class.

To reorder the classes you have to press the button between the forward and back buttons of the knowledge browser (field 2). This button will switch the class tree view between Class List, Domains and Class Tree view (the default view). Please note that button has the name of the view you will switch to (and not the present view). In the Class List mode you will see all classes presented in a syntax showing sub-classes after a dot. So, a sub-class of "Top Goals/Undefined" class is presented as "Top Goals/Undefined.sub-class". You can simply drag and drop the presented class entries in order to manipulate the order of classes and sub-classes. The only class you cannot move is "Top Goals/Undefined".

Please look at the KE Tutorials for more details.

Remarks

- Please realise that when editing a constraint, you will edit this constraint for all relations it is connected to.
- Furthermore, realise that if you do not try to connect possible existing constraints first, you will end up with numerous amount of the same constraints that you cannot delete without destroying projects after you have release your knowledge base. Therefore, when you need a constraint to a relation, first try to connect an existing one and or modify an existing constraint before creating a new one;
- It is also important to realise that renaming of a parameter will automatically modify relations and constraints they are part of.**However, parameters in dataslots and between quotes will not automatically be renamed.** You have to use the **Search & Replace...** option (in the right mouse window) to make sure these parameters are also renamed.
- In addition to all the actions above, please use the right click mouse menu to see all kinds of additional options such as a knowledge base clipboard to copy knowledge.

General User Interface