General Quaestor Graphical User Interface (GUI)

Introduction

This document assumes you are using the latest public Quaestor release. You can check what is this latest release number here.

Realize that the following information is a generic description of the user interface. Specific content of a knowledge based system will determine the specific information it presents. To make a separation between the responsibilities of Quaestor and the developer of the knowledge based system, the Knowledge engineer, think about the following:

1. Names, descriptions, links to documentations, pictures, process order, process elements, requested input and choices, etc. all is the responsibility of the knowledge base developer, the Knowledge engineer within the organisation;
2. The way information is presented, how you are informed during the process, how to provide information, etc. is the responsibility of Quaestor;

Information on the knowledge based system should be mainly provided in the system itself together with external documentation such as a "Getting started" document to get you going.

The Quaestor interface is of a Multiple document interface [MDI] type. This means that it has one main window with several "child" windows inside this window. These Childs are representing separate functionalities of the system, such as, presenting information, browsing through knowledge, working with knowledge, etc.

The standard layout of the Childs in the interface will depend on the type of use (see User levels). The most advanced users (Knowledge Engineers) need more information (thus more windows) than the most basic users. What type of user type/level you are can be checked in Quaestor using the menu options Help>About Quaestor.

In the section 'Standard views' below you will find an overview of standard layouts, connected to a specific user type/level based on the user licence. Moreover, a description of all interface components is given.

1 Start-up of Quaestor

You can start Quaestor in three ways:

1. Double clicking the Quaestor program shortcut;
2. Double clicking the relevant knowledge base;
3. Double clicking the relevant project file;

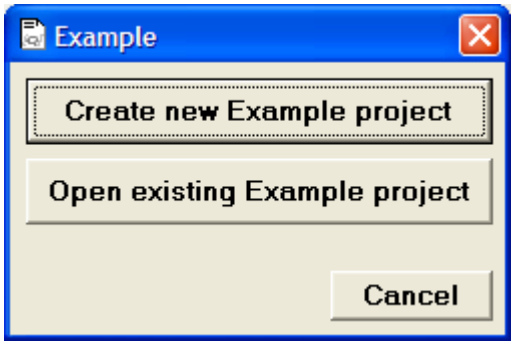
What will follow will depend on the way a knowledge base is protected and the type of user license you have (see also StartQuaestor and User levels for details).

1.1 In general

As mentioned earlier, you can make a distinction between users and developers.

Users (either **End-users** or **Domain Experts**) will **always** work in a project file (*.qpf).

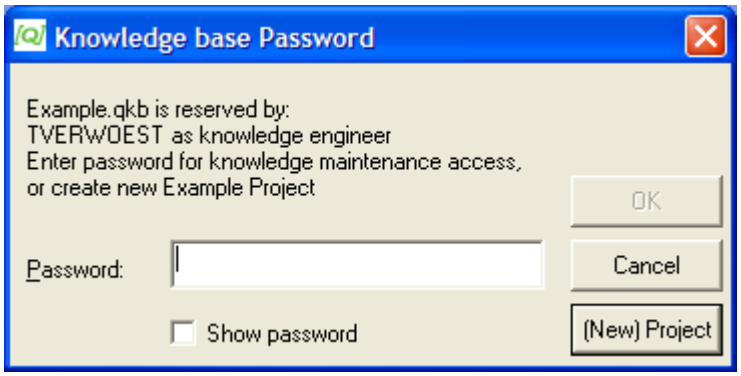
Thus, on starting a knowledge base (*.qkb), these users will always be prompted whether they want to create a new project for a knowledge base, or use an existing one.



Start of the "Example" knowledge base

When you answer to open an existing one, you get a dialog to open an existing project file. When you choose to create a new project, you are able to give a project name and open this project. When you choose Cancel in this dialog, Quaestor will shut down.

Developers (**Knowledge Engineers**) are able to open knowledge bases and therefore will not automatically be requested to create a project. For protected knowledge bases, Knowledge Engineers will be prompted for a password.



Start of the protected "Example" knowledge base as Knowledge Engineer

1.2 Starting as End User

When you have an end-user license, using the Quaestor shortcuts will start Quaestor without any knowledge base or with the knowledge base as defined in the Files tab of the Tools>Options window.

When Quaestor starts without a knowledge base, you have to select either a project or knowledge base through the File menu. When a default knowledge base is given in this Options windows (Tools>Options), Quaestor will start with this default knowledge base followed by the request to create a new project or open an existing project (as discussed above).

When you double click the knowledge base as End-User, you will automatically be prompted to create a new project or open an existing one.

When you double click a project, you will be either requested to provide a password (see also knowledge base protection or the project will open.

Note that in case a project is protected against unauthorized modification, an End-User will automatically open the project read-only.

1.3 Starting as Domain Expert

For a Domain Expert the behavior is mostly comparable to that of an end-user. The only difference is that for a password protected project, a Domain Expert will always be prompted for a password.

1.4 Starting as Knowledge Engineer

For a knowledge engineer, using the Quaestor shortcuts will start Quaestor with an empty knowledge base named Newqkb as the first tree node below the standard Quaestor node in the knowledge browser.

When you double click a specific unprotected knowledge base, Quaestor will start with this knowledge base presenting a node in the tree below the Newqkb node. **Please note that this knowledge base is always protected against deleting frames, so you can still unprotect the knowledge base in the File menu.**

When you open a protected knowledge base, you will be prompted to give the password or the option to start an existing or new project (the same dialog as for the users will than be shown).

Please note that, when you start a project, the view of the GUI will become the same as for normal users.

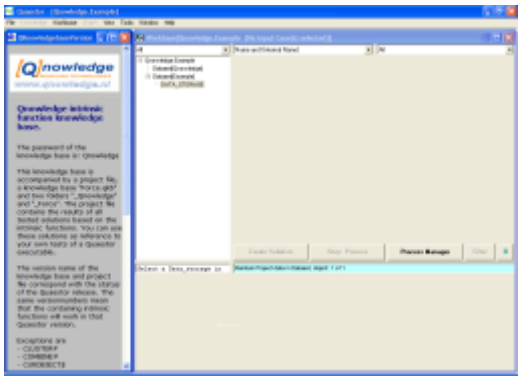
2 Standard views after startup (assumed resolution 1024 x 768)

The standard Quaestor GUI layout can be rearranged at any time by selecting the menu options: View>Window configurations. As with the start-up of Quaestor, in general there are two standard views:

1. As a User, thus when working in a project (for every user level);
2. As a Knowledge Engineer working in a knowledge base;

2.1 Standard view for an End-user, Domain expert and Knowledge Engineer using a project

For an End-User, a Domain Expert and a Knowledge Engineer in a project file, View>Window configurations>User will result in:



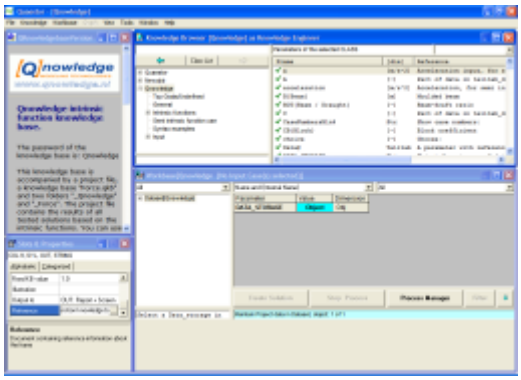
Click here to see large..

Your screen will be divided into an **Explanation window** and a **Workbase**. The **Explanation** window provides all information about the items you select in the **Workbase**. The **Workbase** is your working area to carry out your processes/calculations, provide input and browse through all results.

Please note that specific details of the content inside the **Workbase** and presented in the **Explanation** window will always depend on the knowledge base itself.

2.2 Standard view for a Knowledge Engineer in a knowledge base

View>Window configurations>Knowledge engineer as a knowledge engineer in a knowledge base will result in:



Click here to see large..

Please note that you will only be able to get this layout when you are in a knowledge base. When you are in a project, the user mode will be kept.

As a Knowledge Engineer you will have two more windows in addition to the Explanation window and the Workbase: Knowledge Browser and Properties .

The Knowledge Browser is your main access to the knowledge (such as parameters, relations, constraints) in the system (stored in frames).
The Properties window provides access to specific properties for the knowledge (or frames = parameters, relations, constraints) you are able to select in the Knowledge Browser.

Instead of the Explanation window, which is only for presentation, you also have the Frame Viewer. This Frame Viewer enables you to modify the reference and data slot information for a frame and the dimension of parameter frames. To get the Frame Viewer instead of the Explanation window, select "Use classic frame viewer" in the "Appearance" tab of the Options windows (Tools>Options...).

For more Knowledge Engineer specific information also go to User interface for the Knowledge Engineer.

3 General concept of the Interface for users

Users are interested in using the knowledge in the knowledge base:

- They want to start calculation and modeling processes;
- Make choices and provide input;
- Browse through results;

Start processes

Every process (calculation, task, whatever) starts with the Process Manager. At the start of a project, this window is automatically initiated, you can also activate this window using the**Process Manager** button (when the classic Workbase buttons are used, push or the S button on the right side of the recorder buttons. We advise to use the new Workbase buttons). Depending on the specific knowledge base, the process manager enables you to:

1. Select data (and make copies, select for modifications);
2. Select a task with or without selected data (when no data entry is defined, only the tasks are presented)
3. Start the process.

What follows is a process of providing input and making choices. Every time you provide input or make choices you confirm these by pressing the "Accept input & Continue" button (or the play button, depending on your personal settings, see Workbase buttons).

Choices and input

You give input and choices in the Workbase, the right part of the interface. And while doing so, you are provided with additional information on selected parameters or values in the Explanation window on the left side. The Explanation window will always follow your selection in the Workbase (either a tree item, something in the list or something in the table of the Workbase). **Furthermore, note that Quaestor also provides feedback by means of colors and font styles.**

Using the Workbase buttons in the Workbase, you can start or restart a process, accept choices and input and interrupt or stop a process.

By providing answers and choices to questions in the Workbase and using the Workbase buttons to continue, you will be guided through an analysis, design or engineering process. If the correct input is provided and choices are made, you will end up with a result. Quaestor will inform you if this was not possible.

Browse results

When a result is achieved. You can go through all input, intermediate and end results in the Workbase by selecting values and (in case of documents) double clicking values. Note that associated programs will start and load the associated files when you double click them. A Back to Quaestor window w ill appear on top of the started program in order to return to Quaestor.

Like during the dialog, you will see that the results are presented in all kinds of colors and font styles. These colors and styles have a meaning, go to Qu aestor Workbase colors and font styles to read about the details.

Note that a Knowledge Engineer is able to hide values for users. Thus, whether you see all or a filtered selection of values will depend on the application. A Domain Expert and Knowledge Engineer are always able to unhide the hidden data and values using the **Ctrl+H** key combination.

4 General concept of the Interface for developers

Besides using the knowledge base, the general interest of the developer is to maintain and add knowledge to the knowledge base:

- Browsing through available parameters, relations, constrains, entities, etc.;
- Editing these;
- Test/use the knowledge (see above);

As a result (and mentioned in the above standard views), the knowledge engineer required additional windows, such as the Knowledge Browser, Proper ties and the Frame Viewer to have additional views on the knowledge base.

Browsing knowledge

By means of the Knowledge Browser the KE is able to browse through all parameters, relations and constraints (frames). The Frame Viewer and Proper ties will follow the selections in the Knowledge Browser and enable you to modify all kind of aspects on the selected frames. Furthermore, the Knowledg e Browser will follow your browsing through the Workbase.

Because all the Childs are interconnected you can easily follow the relation between a result and its origin. For instance, when you select a value of a result in the Workbase, the Frame Viewer tells you which relation requested the value and which relation has determined the value. In addition the parameter is in focus in the left bottom side of the Knowledge browser, and when you select this, it will show all the relations the parameter is part of, and by double clicking which relations can determine the parameter.

- Flash demo of a simple knowledge base with Workbase-Knowledge Browser connection: click here....
- Flash demo showing the difference between Workbase-Knowledge Browser connection switched on and off: click here....
- Flash demo of a Taxonomy knowledge base with Workbase-Knowledge Browser connection: click here....

Editing knowledge

Editing you use a combination of the Knowledge Browser, Frame Viewer and Properties.

Note that most of the functionality for the Knowledge Browser is part of the right mouse menu functionality, see also in the following sections.

Modifying parameters

Parameter names can be modified in the Knowledge Browser, simply select the parameter and press F2. The changed name will be propagated through the knowledge base as far a this is recognizable by Quaestor. In other words: **Please note that parameters in templates and between quotes are NOT changed!** The "Search & Replace..." tool automatically starts as soon you have changed a parameter name. This tool allows you to search and replace the old with the new parameter name also in data and reference texts of all frames in the knowledge base. In particular attention should be paid to modify input templates for satellite programs.

Any additional modifications to reference text or attributes and data in the data slot can be done by means of the Frame Viewer and Properties. Put the focus on a parameter in the Knowledge Browser and make your changes in the Frame Viewer and Properties.

Adding parameter can be done in two ways:

1. When a relation is added, the new parameters are added to the knowledge base;
2. Use the right mouse menu option "New Parameter/Function..." in the Knowledge Browser (or the main menu);

For Quaestor it is important to know what type of parameter it is adding. You can define this type in two ways:

1. Implicitly, using the pre-defined syntax (see also QuaestorSyntax):
    a. $ at the end of the name means **string type**: the parameter is used for text values;
    b. # at the end of the name means **TeLiTab type**: this is the Quaestor data structure for a combination of Text, List values and Table values.
    c. % at the end of the name means **integer type**: See [1].
    d. Normal parameter name is assumed to be of the **value type**: this can be any real number.
2. By defining this in the "New Parameter/Function..." window;

In other words, when you add parameters by means of a relation, you can also define the type of the parameter.

Note that due to some choices in design, a string parameter in Quaestor cannot be changed to a value type (so removing the $ will not help...). In that case you have to create a new parameter with the correct type.

Modifying relations

Modification of expressions (relations or constraints) is done by means of the Expression Editor. The window automatically opens when you select an expression and press F2. You can see expressions in the Knowledge Browser by:

- Double clicking on the class in the tree view;
- Go to a specific relation through the browsing process described above;
- Make a specific view by means of the filters in the Knowledge Browser;

When the Expression Editor is opened, it shows the expression in the top part and any additional attributes and data in the bottom part. You can simply modify the expression in the normal way.

The Expression Editor contains syntax information through the Tooltip [2] and a proposing mechanism (a drop down box) for available functions and parameters, based on the parameter type of the left side parameter and any further requirements inside a syntax of a intrinsic function. These two assistants work dynamically while typing your expression. Furthermore, we use syntax coloring for string information (not string parameters), brackets and logical words like AND, THEN, ELSEIF, etc.

New relations can be added using the right mouse menu of the Knowledge Browser. The same Expression Editor is opened and the last parameter or relation that was in focus is included as starting point.

Note that, because Qnowledge wants maximum flexibility, realize that you may introduce as many relations as you like. Also while they are all the same! This is not a smart thing to do because, for one thing, you will not know which relations is used in your solution. Furthermore, as mentioned below, deleting the expression is possible but only safe when no projects are based on the knowledge base yet. Thus, be careful with existing and large knowledge bases that are in use by a lot of users.

Modifying constraints

As suggested above, modifying constraints is the same as modifying relations. The only difference is the syntax itself, which should contain logical expression (<,>,=,AND, OR, XOR, etc.).

Please realize that modifying a constraint means modifying this constraint for every relation it is connected to!

Adding constraints is something different. The whole idea of a constraint is that it limits the use of a relation. Therefore, adding constraints (or connecting / removing existing ones) is done by selecting the relevant relation and choose the relevant option in the Constraint sub-menu.

As mentioned above, you are able to make many equal constraints with ease. However, most of the time this is not what you want. For one thing, it will become harder and harder to know which constraint is connected to a relation (for instance, when you want to modify this constraint). **Therefore, we want to stress that you should always try to connect (and sometimes modify) existing constraints.** Think about your requirements. This will save you a lot of time searching for bugs in you knowledge base.

Deleting frames

As already suggested above, deleting frames is not a problem as long as nobody is using your knowledge base. However, due to choices in the design of Quaestor, deleting a frame will result in to renumbering of all the existing frames in the knowledge base, and thus reshuffling the relations between parameters, etc. Solutions in projects based on a previous version of such a modified knowledge base will end up with corrupt solutions as a result.

Note that this problem only occurs when deleting frames. When you simply recycle obsolete parameters and relations for new functionality there will be no problem at all.

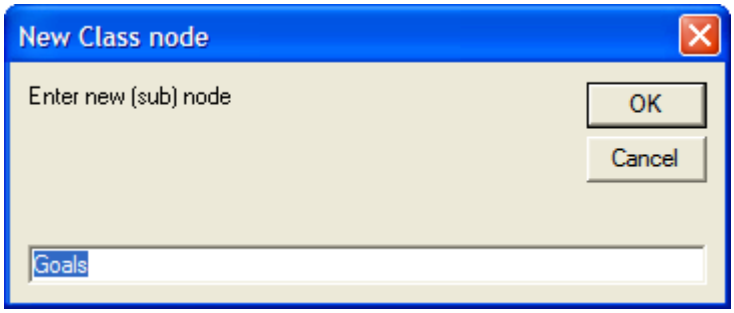
Because we want you to think about this, deleting is normally disabled. You can only enable deleting by selecting the File menu item "Allow deleting frames". Every knowledge base is (re-)protected when you (re)start it in Quaestor.

Organise in classes

Beside a description of all your parameters, relations constraints etc. 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….



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 and Documentation of Knowledge for more details.

Testing knowledge

Every new knowledge component can be tested right away. Simply double click the parameter to be calculated in Knowledge Browser to indicate it is a top goal and press the Create Solution or play button to start your calculation process.

Functionally, the testing is completely the same as described for normal users above (please read this part too). However, presentation of parameter names etc. might differ. Therefore, we advise to do thorough testing in a project with the correct user level (you can modify your level in the Options window (Tools>Options...)).

5 Overview of all window types

All window types are given in the Quaestor View menu. It will depend on your user rights, position in the interface and actions taken by Quaestor, whether these windows are enabled. When any of the windows has disappeared (because another window has covered it), you can re-show them by selecting it in the View menu.

The standard windows are activated when a knowledge base is loaded (depending on the standard view above). The standard windows are:

- Workbase, to provide input during a dialog and navigate through all solutions;
- Explanation, to present all reference information (and all other binary data connected to parameter, objects and relations that can be visualized in a web browser);
- Knowledge Browser, to add, modify and navigate through all knowledge;
- Frame Viewer, to view and change details on parameters, relations and constraints;
- Properties, to view and change properties of parameters, relations and constraints;

In addition to these windows you have windows that are activated by Quaestor when you carry out a specific action, such as the Knowledgebase Clipboard, but for instance also the Web browser to provide additional help for intrinsic Quaestor functions. The additional windows are:

- Trace Viewer, specialized window to debug modeling and evaluation of solutions;
- Web Browser, to view web pages, pdf documents, text, etc. in a separate window;
- Graph, simple viewer for multi-case values;
- Knowledgebase Clipboard, to export knowledge from the Knowledge Browser to a separate file, etc.;
- Workbase Clipboard, to export results from the Workbase to a separate file, etc.;
- Expression Editor, to edit expressions (relations, functions, constraints);

- Compose Text, to compose text based on text fragments in the Knowledge Browser or Workbase;

Finally you have special modal windows, for instance required to control Quaestor in combination with other programs. The special modal windows [3] a re:

- Report, to view reports generated by Quaestor;
- Process Manager, to select datasets and tasks and start a analysis, design and engineering process;
- Back to Quaestor window, enabling and forcing you to return to Quaestor in the correct way;

6 Overview of menu items

Quaestor has two types of menus:

1. The standard menu on the top of the Quaestor [GUI];
2. The right mouse menu which can be initiated in each window when by pressing the right mouse button ( ) showing specific actions related to that window.

A detailed description of all items is presented below.

Please note that it will depend on all kind of aspects whether menu items are enabled. Moreover, some of the menu items may change their name on the basis of the selected context.

For the standard menu items on top op Quaestor, below all items are described. For the right mouse menu items, it will depend on the selection (and user rights) what kind of items are enabled. Only de relevant items corresponding with the selection are presented.

The separation lines in the menu's are represented with "-".

6.1 Main menu items

- File
  - New
  - Open...
  - Open Project...
  - Find Project...
  - Project History...
  - Close KB / Close Project
  - -
  - Save KB / Save Project
  - Save KB As... / Save Project As
  - Save KB as Project...
  - Save Frames as List...
  - Allow deleting frames
  - Protect File... / Unprotect File
  - Encrypt File... / Decrypt File...
  - Clean Sub Folders...
  - -
  - Print Setup...
  - -
  - MRU list
  - -
  - Exit

- Knowledge
  - New Relation...
  - New Parameter/Function...
  - Insert Knowledge base...
  - -
  - Class copy
  - Reset LastChanged to Now

- Workbase
  - Start Dialogue/Text on Workbase buttons
  - Interrupt Dialogue/Text on Workbase buttons
  - End Dialogue/Text on Workbase buttons
  - Process Manager
  - -
  - Make Report...

- Graph
  - Save As...
  - -
  - Copy
  - -
  - 3D Bar
  - 2D Bar

    - 3D Line
    - 2D Line
    - 3D Area
    - 2D Area
    - 3D Step
    - 2D Step
    - 2D Pie
    - 2D XY
- View
    - Knowledge Browser
    - Workbase
    - Frame Viewer
    - Properties
    - Explanation
    - Trace Viewer
    - Web Browser
    - Graph
    - -
    - Knowledgebase Clipboard
    - Workbase Clipboard
    - Expression Editor
    - Report
    - Compose Text
- Tools
    - Check Relations against Values
    - Verify Network Integrity
    - Knowledge base Summary...
    - RTF Template Check
    - Compare Kernel KB's
    - -
    - Options...
    - Chance License... or Register Quaestor Now...
- Window
    - Standard
    - Cascade
    - Tile Horizontally
    - Tile Vertically
    - Arrange Icons
    - -
    - Open Knowledge bases...
- Help
    - Search...
    - Show Knowledgebase documentation
    - -
    - My [Q]nowledge
    - Quaestor.org
    - -
    - About Quaestor...
    - Show Current Users

6.2 Right mouse menu items

In the knowledge base:

The Knowledge browser menu when selecting a tree item will differ from the same action in the Frame information part.

Selection in the tree item:

- Open KB
- Open Project
- Rename KB
- -
- List Parameters
- List Expressions
- -
- Cancel

Selection in the frame information part:

- Select as Goal
- -
- New Relation
- Constraint>

    - Add New...
    - Connect...
    - Disconnect...
- New Parameter/Function
- Delete Frame
- Rename or Edit (depending on selection of a parameter or relation)
- Include Binary in frame
- -
- Frame to Clipboard
- Set to Clipboard
- Parameter to Dataset
- -
- Properties
- Illustration
- Complete expressions
- Compressed expressions
- -
- Add to Watch list
- Protect Input
- Summarize Relation
- List all values in Project
- Compose Text
- Search & Replace
- Expression Evaluator
- -
- Cancel

In the Workbase:

For the Workbase the available options in the right mouse menu will depend on:

1. the type of knowledge base (Classic or Taxonomy-based);
2. whether you are in an active or inactive Solution;
3. what you have selected in the Workbase;

In particular the Workbase right mouse sub-menus will contain different options in these situations.

Due to the above situations (use cases) menu items might be different or disabled. Below we give all menu and sub-menu items grouped according to the most common use cases.

Main menu items

The **main menu** of the Workbase contains the following options:

- Workbase main menu
    - Refresh
    - Unreject
    - Copy values
    - Paste values
    - -
    - Solution/Object/Taxonomy/Entity sub menu))
    - Database Input
    - Process Manager
    - Value to Table
    - Value to List
    - Unfold
    - -
    - Selection to Clipboard
    - All to Clipboard
    - -
    - Filter Values
    - Make Report
    - Make Function
    - Make Polynome
    - Show Cluster
    - -
    - Cancel

The Workbase sub menu covers the following prime use cases as listed below.

During running or examining a Classic solution (not all options are visible or enabled at the same time):

- Solution
    - New
    - Accept all proposals
    - Copy
    - Delete
    - Delete Object

    - Recalculate
    - Strip
    - Trace On
    - ((WorkbaseFind|Find)
    - ((WorkbaseAddToCluster|Add to Cluster)
    - ((WorkbaseMutate|Mutate)
    - ((WorkbaseExpand all|Expand all)
    - -
    - Transpose Cases
    - Case Matrix
    - Add Case
    - Insert Case
    - Delete Case(s
    - Delete Parameter(s
    - Compress All
    - -
    - Show no Defaults
    - Compose Text Value..
    - Include Binary Data..
    - Save as Initial Values

During engineering in a Taxonomy (not all options are visible or enabled at the same time):

- Taxonomy
    - Delete “EntityName”
    - Include New Entity
    - Create Relation
    - Value = Input
    - Edit Relation
    - Show related Entities
    - ((WorkbaseInstantiatePar|Instantiate “Parameter”)
    - -
    - Transpose Cases
    - Case Matrix
    - Add Case
    - Insert Case
    - Delete Case(s
    - Delete Parameter(s
    - Compress All
    - -
    - Show no Defaults
    - Compose Text Value..
    - Include Binary Data..
    - Save as Initial Values

During running or examining a Taxonomy solution (not all options are visible or enabled at the same time):

- "EntityName" (the actual name of the selected Entity)
    - Copy
    - Delete
    - Synchronise with Taxonomy
    - Recalculate
    - Delete EntityName
    - Recalculate Branch
    - Trace On
    - ((WorkbaseFind|Find)
    - ((WorkbaseMutate|Mutate)
    - ((WorkbaseExpand all|Expand all)
    - -
    - Transpose Cases
    - Case Matrix
    - Add Case
    - Insert Case
    - Delete Case(s
    - Delete Parameter(s
    - Compress All
    - -
    - Show no Defaults
    - Compose Text Value..
    - Include Binary Data..
    - Save as Initial Values

In the Graph window:

- Workbase Graph
    - Save As...
    - -
    - Copy

    - -
    - 3D Bar
    - 2D Bar
    - 3D Line
    - 2D Line
    - 3D Area
    - 2D Area
    - 3D Step
    - 2D Step
    - 2D Pie
    - 2D XY
    - -
    - Cancel

Other menu's:

- Other right click menus
    - Undo
    - Cut
    - Copy
    - Paste
    - Delete