

Tutorial 6: Evaluation and Reporting

Tutorial 6 Evaluation and reporting	Learning goals 1. Using the INCASE, THEN and ELSE functions. 2. Using the SORT# function. 3. Creating reports using the WINWORD\$ function.	Prior knowledge 1. Interface 2. Tutorial 1: Getting Started 3. Tutorial 2: TeLiTabs and Objects 4. Tutorial 3: Objects, interpolation and integration 5. Tutorial 4: Telitabs and solution management 6. Tutorial 5: Satellite programs	
--	--	--	---

1 Objective

In this tutorial we will evaluate the systematic variations of the design using a weighted average of all parameters describing the compatibility of the ships. Then we'll sort the designs in order of compatibility and finally a report is created in MS Word for each design.

Start

To start this tutorial, a knowledgebase is available [\[Tutorial 6 start.qkb\]](#). This knowledgebase contains the results of tutorial 5 and is the basis for this tutorial.

Also a Word template file ([Report.rtf](#)) has been provided. Make sure you save this file in the Applic directory of your knowledge base so it can be found during calculation.

2 Create a new Dataset as starting point

Create a new object *Ship_evaluation*. Add this to the list of the present *Data* object in Dataset (you can either leave the existing *Shipsobject* or delete it). Make sure that the Determined by property of *Ship_evaluation* is OBJ, otherwise you will discover later on that the data inside the object is not taken to the solution.

1. Add the following parameters to the knowledge base:

- *Stab*;
- *Comf*;
- *Rt*;
- *PE*;

All parameters are values and the "Determined by" of Slots & Properties should be *OBJ* because they should be taken from the dataset. *Stab* and *Comf* are without a dimension. *Rt* is in [N] *PE* is in [W]. Move *Stab* and *Comf* to the *Stability* class, *Rt* and *PE* to the *Resistance* class.

2. Now add data to the dataset that we are going to use during the calculations:

- Select the *Ship_calc#* parameter in the *Ship_calc#* solution en press F2 to see the [TeLiTab](#) as ASCII data in the editor.
- Select and copy all data.
- Go to *Ship_evaluation* in the *Data* object of the Dataset, right mouse click and select "Database Input..."
- Select editor, paste the selected data and press ok;
- Add all parameters as new;

This is the starting point of our evaluation.

3 Evaluation of the designs

Create the following evaluation relations:

- $$\text{Efficiency} = 1 + 25.7 - (\text{PE} / (\text{DISP} * \text{V}_{\text{eco}}) * 2.25)$$
- $$\text{Journey} = 1 + (28 - (10000 / \text{V}_{\text{eco}} / 24)) * 2.25$$
- $$\text{Grade} = \text{INCASE}(\text{GM}, \text{LT}, 0.4, \text{THEN}, 0, \text{ELSE}, (\text{Stab} + \text{Comf} + 1.2 * \text{Journey} + 1.2 * \text{Efficiency}) / 4.4)$$

Add the evaluation parameters as goals to the *Ship_evaluation* object using a relation:

- Sort the ships, based on their grade

- To check whether all this works, carry out a calculation. Start the process manager. Select Data and select Sorted_evaluation#. The content of Sorted_evaluation# will be:

4 Check the template file of the report, create a report

This template should always first be checked on strange RTF code. Use the menu item Tools>RTF Template Check... You will be asked to select a file and the Template delimiter (usually #, as in our file). Then Quaestor will check, provide a log file and store the original and changed template. The log is as follows:

Do this once or twice until you only see sensible template code in the log.

We will end this tutorial creating a report of each ship. Add the following relation to create the report:

Report\$= WINWORDS\$ Report.rtf("Report.doc",0,Ship no,Lpp,B,T,DISP,GM,Stab,Comf,V eco,Rt,PE,Efficiency,Journey,Grade)

The **WINWORDS()** function will create Word files named *Report*.doc.* in your report directory and returns the name of this file to **Report\$**. In multiple case situations a number (*) is added to the name of the document.

So make sure the reports are created, modify Evaluation# and add Report\$:

Evaluation#:=Ship_evaluation(@Journey, @Efficiency, @Grade, @Report\$)

Make sure Report.rtf can be found.

To generated the total solution, select Data in the dataset and run Evaluation# as top goal. A report is generated for each ship and added to each case. If parameters are requested during the calculation, check the following:

- Does Ship_evaluation have the OBJ Determined by property?
- Do the parameters carrying out intermediate values have the correct Determined by properties?

You can open a document by double-clicking it in the object:

[blocked URL](#)

One of the Word reports:

[blocked URL](#)

Final remarks

These six tutorials showed both the way to create a knowledge base and some basic elements about working with ranges, objects, data and external processes, in general used in [classical knowledge based applications](#) and [Scenario base applications](#).

Please note that it is possible to create the same functionality in many different ways. So feel free to experiment. For instance, it is possible to make the ship design variations (B, Lpp, etc.) part of the calculation process enabling a user to vary the design space he or she wants to investigate. Moreover, we did not discuss the [SCENARIO\\$\(\)](#) function to create a scenario of input and output actions or all kind of other [intrinsic](#) functions.

Moreover, we have not discussed any specific [knowledge engineering](#) related to [Taxonomy based applications](#). For the latter we will introduce a new set of tutorials.

For more inside information you can use the following KE resources on Quaestor:

- [Overview of the general Quaestor syntax](#)
- [Overview of all intrinsic Quaestor functions](#)
- [Overview of all Quaestor attributes](#)
- [Overview of all Quaestor constants and reserved parameters](#)
- [Overview of all Quaestor dimensions](#)

Also take a look at examples, these are always very helpful to solve a particular problem. [\[Example applications\]](#)

5 Check

You can verify your results by comparing it to [\[Tutorial 6 finish\]](#)

[<< Back to tutorial 5 Overview](#)