

# **The Graphical User Interface for CAPRI**

## **version 2009**

**Wolfgang Britz**  
**Institute for food and resource economics**  
**University of Bonn**

**Bonn, May 2009**

## Acknowledgments

Many people have over the years contributed to the development, maintenance and application of the CAPRI modelling system, and after now ten years since a first prototype was constructed, it is almost impossible to list them all and name their specific contributions. The author opted for this rather technical paper to refrain from citing the different (working) papers which shed more light on methodological questions, but rather refers in general to the CAPRI documentation.

Nevertheless, it is only fair to mention Hans-Josef Greuel and Andrea Zintl, who both long before CAPRI was born have already developed software concepts and code which underlined to a large extent until 2006 the DBMS of CAPRI, and in parts, its Graphical User Interface. Finally, Alexander Gocht over last years contributed to the Java code underlying the GUI.

The work described in here would have been impossible without the funds by different donors, mainly the EU Commission.

All errors in text and code remain with the author.

## The author

Dr. Wolfgang Britz is a senior researcher and lecturer with the Institute for Food and Resource Economics at the University of Bonn, and has co-ordinated since several years the activities based on the CAPRI modelling system. Its responsibilities further on include the methodological concept of CAPRI and , to a larger extent, its software implementation.

### Contact:

Dr. Wolfgang Britz

Institute for Food and Resource Economics, University Bonn

Nussallee 21

D-53115 Bonn

Tel.: ++49-(0)-228-732502

Wolfgang.Britz@ilr.uni-bonn.de

# Content

Background .....	9
Initialization .....	10
Logging in .....	10
Choosing a initialization files.....	10
Linking the GUI to the local CAPRI installation .....	11
GAMS settings .....	11
Getting help .....	13
Basic layout of the GUI.....	14
The different work steps.....	16
Build database .....	16
The work step “Generate baseline” .....	19
The task run simulation .....	21
Define scenario task .....	22
Run simulation task .....	23
The task: Exploit scenario results.....	26
The task: Collect meta information .....	27
Batch execution .....	29
Task Generate GAMS documentation.....	31
Exploitation of.gdx-files.....	32
An example: loading data from the spatial downscaling.....	33
Interaction with GAMS .....	36
Exploitation tools .....	37
The structure of the GAMS generated.gdx files.....	37
Loading the data from.gdx files .....	37

Multi-dimensional viewer with pivoting and exporting possibilities .....	38
Pre-defined views .....	38
View selection .....	38
Navigating in the outer dimensions of the viewport.....	39
Column and row selection .....	39
Predefined selection groups.....	40
Selection of the view type .....	41
Manually changing the pivot.....	41
Changing view options.....	42
Showing a histogram windows.....	44
Working with tables .....	45
The toolbar .....	45
Tooltips for column and row headers.....	46
Drill-down .....	46
Clipboard export.....	46
Export to file.....	46
Sorting .....	47
Numerical filtering based on cell content.....	47
Changing the row height and column width with the mouse .....	47
Adding statistics .....	48
Outlier detection algorithms implemented .....	50
Working with graphics .....	52
General handling of graphs .....	52
“Walking” through the data.....	55
Exporting the graphic to file.....	55
Exporting the graphic to clipboard.....	56

Bar charts.....	56
Line and point charts .....	57
Pie charts .....	59
Spiderweb plots .....	60
Box and Whisker charts .....	61
Histograms .....	62
Markov charts.....	63
Flow maps .....	64
Pie chart maps .....	66
Colored thematic maps.....	66
Changing the classification and the legend .....	67
Adding a histogram window to a map.....	68
Shrinking polygons according to UAA share.....	69
Area weighted classification .....	70
Excluding zeros from classification and removing small and large values.....	70
Classification method .....	70
Integration distribution information in the map window.....	73
Color table .....	73
Changing the way the legend is drawn.....	77
Copying the map to the clipboard or saving to disk.....	79
Changing the title of the map .....	79
Zooming in and out and navigating in the map .....	79
Getting data for specific polygons.....	80
Highlighting specific regions in the map.....	81
Updating the map .....	85
Adding region label to the map .....	85

Storing and re-loading your settings.....	86
Exporting the data underlying the map .....	87
What are the HSMUs and what information is available? .....	99
How to visualize the HSMU information.....	99
Loading data based on the GUI presets .....	99
Examples .....	103
Drawing a map showing the nitrate surplus for EU27 at regional level in the base year.....	103
Drawing a map of the High Nature Value Farmland indicator for Belgium for the base year.....	105
Drawing a map showing changes between the base year and the ex-ante results from the baseline	109
Drawing a map with the base year results next to one showing changes from the base year to the baseline results .....	112
The software behind the mapping viewer and the CAPRI exploitation tools .....	115
CAPRI tasks as business model .....	116
Execution of tasks via a GamsStarter and GamsThread.....	116
Refactoring the mapping part.....	117
Views as the basic concept.....	118
Data model .....	118
Client based solution .....	119
The geometry model.....	119
The structure of the XML definition files for the views.....	121
General comments.....	121
Necessary tags for tables .....	121
Defining the items of the table .....	122
Additional tags .....	123
<defpivot> .....	123
<defview>.....	124
<COO> .....	124

Alternative texts for the dimensions ..... 124

Filters for the elements in the different dimensions ..... 125

Attaching long texts and filters to elements ..... 125

Index..... 127



## Background

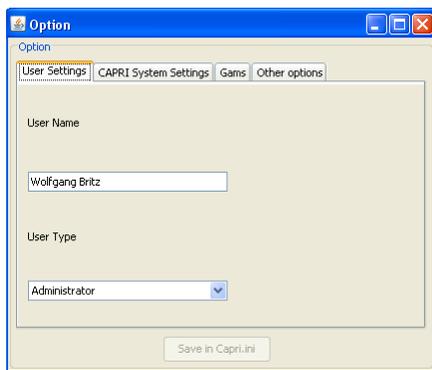
The use of the CAPRI modelling system is increasing, and the user group becomes more diversified. Whereas in the first years, almost all users had directly contributed to model development and were familiar with the underlying GAMS code, more and more users now get to know about the system during training sessions, and have only a limited knowledge of GAMS and the CAPRI GAMS code. Already a few years back, a Graphical User Interface (GUI) was developed in order to support users to apply CAPRI for simulations and exploit results. For reasons laid down further down in a short chapter, this GUI needs were now revised in major parts. That paper both explains the usage of the new GUI as well as the underlying software concept. It is structured as follows. The first chapter gives a short overview over the different work steps necessary to finally allow simulations with CAPRI.

The new GUI is as the old one realized in Java. The small technical background paper "Refactoring of the CAPRI GUI and integration of new functionalities" informs about the underlying software design.

# Initialization

## Logging in

The first step when the CAPRI GUI is opened for the first time is to set the user name and level, which is done by selecting the user menu from the menu bar. As long as no user name is entered, the user cannot change its type and will only have exploitation rights. The user type “runner” has additionally the right to run scenarios. A user of type “administrator” can perform all operations, including generation of a new data base and calibration of the modelling system. In order to access the user settings, choose

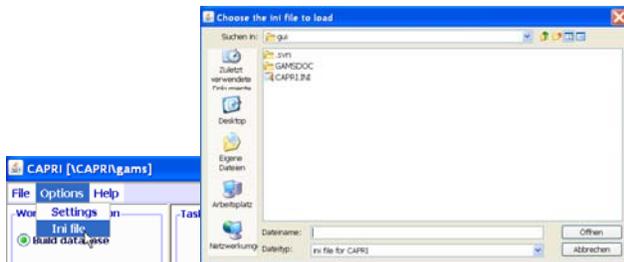


The user and user types can also be seen in the bottom panel of the GUI:



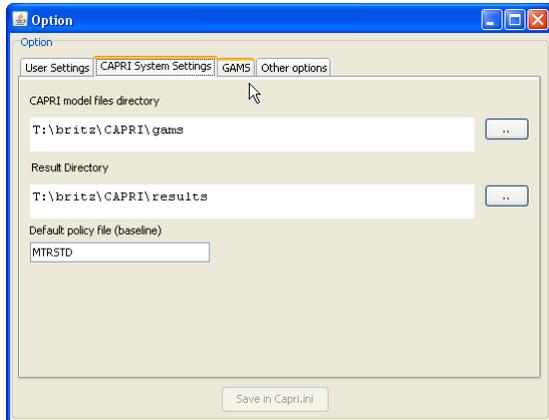
## Choosing a initialization files

Some user require to have several CAPRI versions installed in parallel. In order to ease the task, the user can call the GUI with a specific ini-file by defining appropriate batch command file. Alternatively, the ini file can be changed via the options menu. Each ini-file may then point to different directories, according to the settings discussed in the following.



## Linking the GUI to the local CAPRI installation

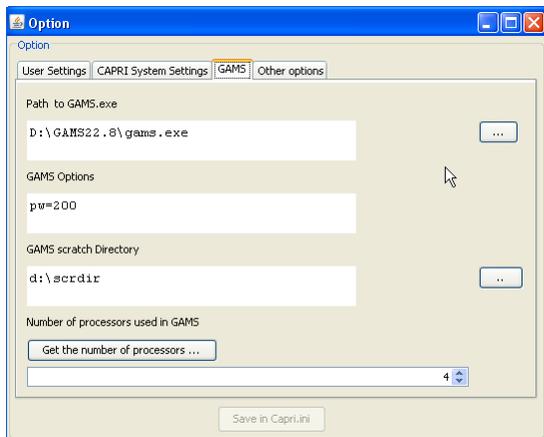
Next, the GUI needs to know where your CAPRI system is installed.



The “CAPRI model files directory” points to the location of the GAMS sources for CAPRI whereas the “Result directory” points to the location where results from CAPRI tasks will be read from and written from. Changing these settings allows switching between different installations for advanced users, e.g. when different branches from the CAPRI software versioning system are installed.

## GAMS settings

In order to generate results, a GAMS installation and license are required. The relevant settings are found on the “GAMS” tab:

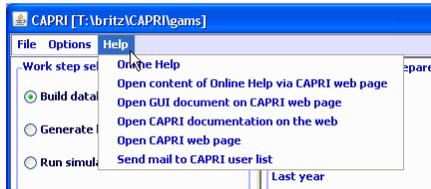


The “Path to Gams.exe” points to the actual GAMS engine to use. Currently, versions 22.3 and higher are supported. It is recommended to use GAMS 22.6 and above to benefit from the grid solve facility. The grid solve facility will use up to “Number of processor used in Gams” to solve model instances in parallel. The button “get the number of processors ...” will retrieve the number of available processors in the computer.

The “Scratch Directory” will be passed to GAMS and determines where GAMS stores temporary files. A directory on a local disk (not one on a file server) should be chosen.

The “GAMS options” field allows the user to send its own settings to GAMS, e.g. as shown above the page width use in listings.

## Getting help



The “Help menu” allows opening the online help system, which can be invoked by pressing “F1”. A copy of the content is also stored on the CAPRI web page and can be accessed via the second menu item. “Open GUI document on CAPRI web page” will open the current document.

## Basic layout of the GUI

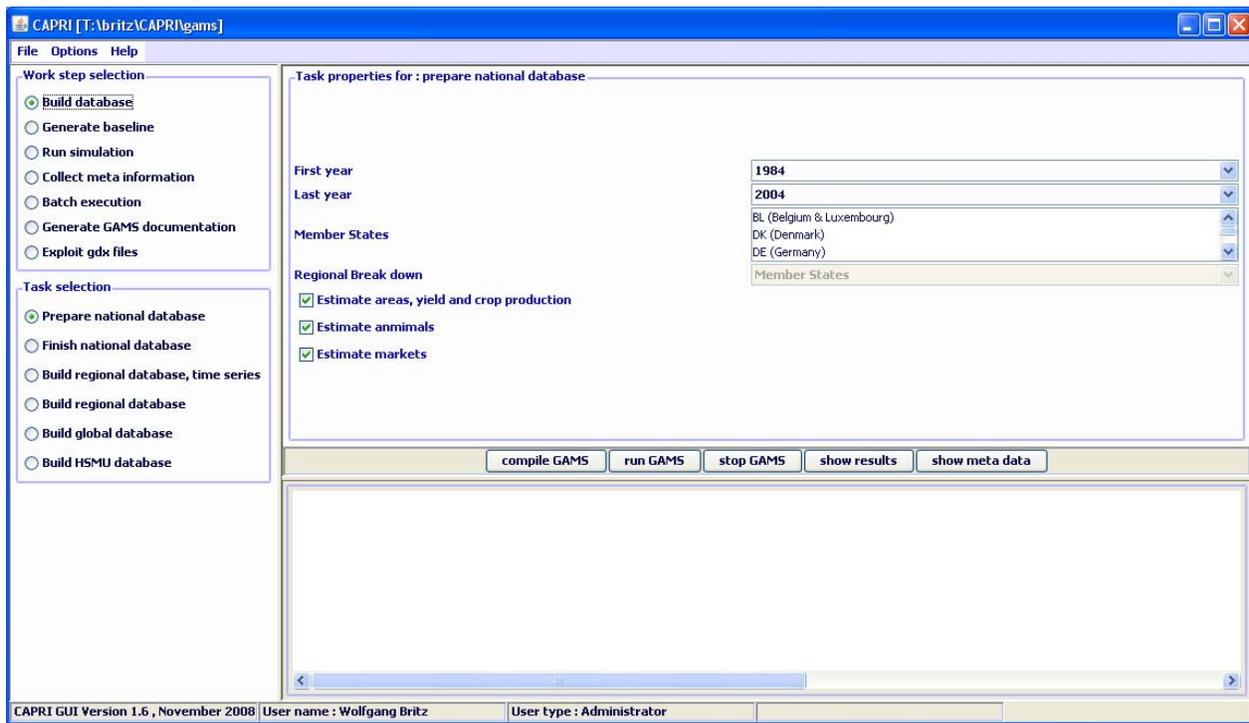
The GUI is generally structured as seen below. The left upper hand panel allows the selection of the different CAPRI work steps. The left lower hand panel lists the tasks belonging to the work steps. In both cases, only one selection is allowed. The right hand side offers controls depending on the properties of the task. There are buttons allowing starting the task, and typically a window which collects information at runtime. The footer lists the user name and type, and comprises progress bar.

For tasks linked to a GAMS program, the buttons as shown below will be active:

- “compile GAMS”: starts the GAMS compiler, but does not execute the program. A listing file will be generated. Used to test if a program compiles without errors.
- “run GAMS”: tries to execute the GAMS program. A listing file will be generate where possible compilation or run-time errors are reported.
- “stop GAMS”: sends a “signal interrupt” to the GAMS engine. It may take a while until GAMS reacts and stops with an error message after running its finalization routines.
- “show results”: loads the results form the task in the CAPRI exploitation tools
- “show meta data”: loads the meta data for the results from the task in the CAPRI exploitation tools.

*Note:* for exploiters, the three buttons referring to GAMS will be not visible. The same holds for runners and the work steps “Build data base” and “Generate baseline”.

## Graph: Basic layout of the GUI



## The different work steps

Each work step may comprise different tasks. No task will require starting more than one GAMS program, but some tasks will start the very same GAMS program with different settings. Some tasks will not start GAMS, but other tools inside the GUI.

The different work steps are shown in a panel in the lower left corner of the GUI, and are presented by so-called radio-buttons, which means, that only one button can be selected at any time.

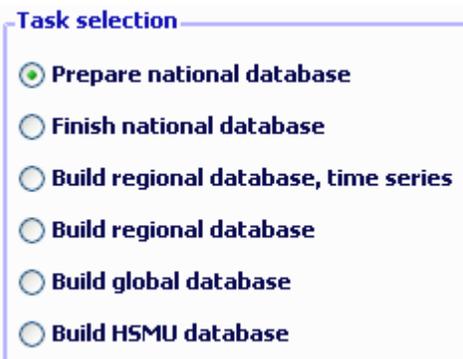
### Graph: the work step panel



Each work step may comprise several tasks, which are shown in second panel below the work step panel. The content of the panel hence changes when the user selects a different work step. Again, the different task panels comprise radio buttons for selections purposes.

### ***Build database***

#### Graph: the task panel for “build database”



Building up the data base is the logical starting point in sequences of work steps. A new data base for the model needs to be constructed either after updates of the underlying statistical raw data, or after methodological changes in the code affecting content and structure of the data base. Controlling if the different tasks necessary when updating the model yielded satisfactory results is a time demanding task which requires in-depth knowledge about the quality of the different in-going data and the logical relations between the different elements of the data base. Users interested in ex-ante policy analysis are usually better off by taking the data base as given, and consequently, the work step is disabled for users which have no “administrator” status.

The work step consists of six different tasks:

1. **Prepare national data base:** *Generation of complete and consistent time series at national level, mainly based on Eurostat data (CoCo, from Complete & Consistent).* CoCo runs per Member State simultaneously for all years, if data from other Member States are used to derive fallbacks as EU average, only the raw statistical data are used. The user can only choose which countries to run, and which years to cover.
2. **Finish national data base:** *Completion of the CoCo data by time series on consumer prices and certain feeding stuffs.* In both cases, it turned out that only the complete and consistent time series for all Member States from 1. provide a good basis for that step. The step is hence run simultaneously for all Member States and years, based on the results of the CoCo task. Here, only the years to cover can be chosen by the user.
3. **Build regional data base, time series,** *Generation of time series at regional level (CAPREG).* The treatment of years in CAPREG is not identical. For all years, activity levels, output coefficients and input coefficients (excluding feed inputs) are generated. However, only for the base period, a three year weighted average around the chosen base year, feed input coefficients are estimated and the supply models are calibrated based on techniques borrowed from Positive Mathematical Programming. The user can hence choose for which Member States to run CAPREG, for which years and for which base year. Equally, the farm type module may be switched on or off.
4. **Build regional data base (CAPREG).** Currently the same as three, only that the base year data will be loaded instead of time series.
5. **Build global data base (GLOBAL):** *Building up the international data base.* The step includes aggregation of Supply Utilization Accounts and bilateral trade flow matrices from FAO to the product and country definitions of CAPRI, aggregation of the supply and demand elasticities from the World Food Model to the product and country, estimation of bi-lateral transport costs and conversion of the FAPRI baseline to the product and regional aggregation of CAPRI.

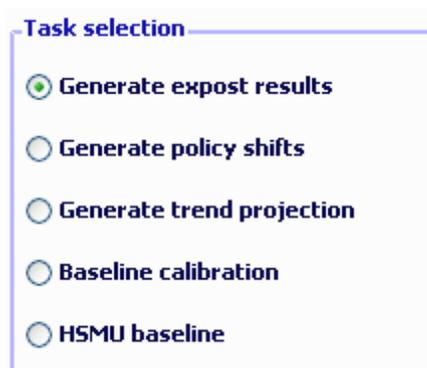
**6. Build HSMU data base (CAPDIS\_GRID):** spatial downscaling of regional results for the base year to 1x1 km grid cells.

The underlying methodology for the different work steps is described in detail in the CAPRI model documentation. The sequence of the tasks as described above follows the work flows. It should be mentioned that certain preparatory steps, as downloading updated data from EuroStat, and converting these data into GAMS tables read by CoCo and CAPREG are no yet integrated in the GUI.

The actual controls available will depend on task. Please use the “F1” button to open the online help to get detailed information on settings for the tasks.

## The work step “Generate baseline”

Graph: the task panel for “Generate baseline”



For manifold reasons discussed in methodological papers, economic models as CAPRI are not suited for projections, but as tools for counterfactual analysis against an existing comparison point or an existing set of ex-ante time series. The point in time or these time series are called “base line” or “reference run”. CAPRI “runners” which use the model for ex-ante policy simulation do not need to construct their own baseline, but are typically better off by sticking to the baseline provided on a yearly basis along with the latest version of the GAMS code, data base and software. Accordingly, the step and the included tasks are only for user type “administrator”. According to current planning, the baseline will be updated in close co-operation with DG-AGRI twice a year in early summer and early winter, following the release of a new “medium term market outlook” by DG-AGRI.

The CAPRI baseline is a mix of trends, expert knowledge and automated checks for logical consistency, and is constructed by a sequences of tasks:

1. *Generation of ex-post results.* Albeit not strictly necessary for the base line, the ex –post results often prove quite helpful when analysing the reference run. The ex-post results are model run for the base at base year policy and other exogenous parameters, inflated to the chosen simulation year.
2. *Generation of policy shifts.* In order to capture in the later trend projection the effect of policy changes between the base and the simulation year, so-called policy shifts are calculated by applying the ex-ante policy to the state-of-world (technical progress, management practises, population numbers, per capita income, consumption patterns etc.) in the base year. The policy shifts are defined as the relative changes against the base year of implementing the ex-ante policy.
3. *Generation of the trend projection.* The trend projection task is rather time consuming, and may run several days when the farm types are included. It consists of several sub-tasks. Firstly,

4. *Baseline calibration.* In the final task, the results from the trend projection serve as the major input to generate the baseline.

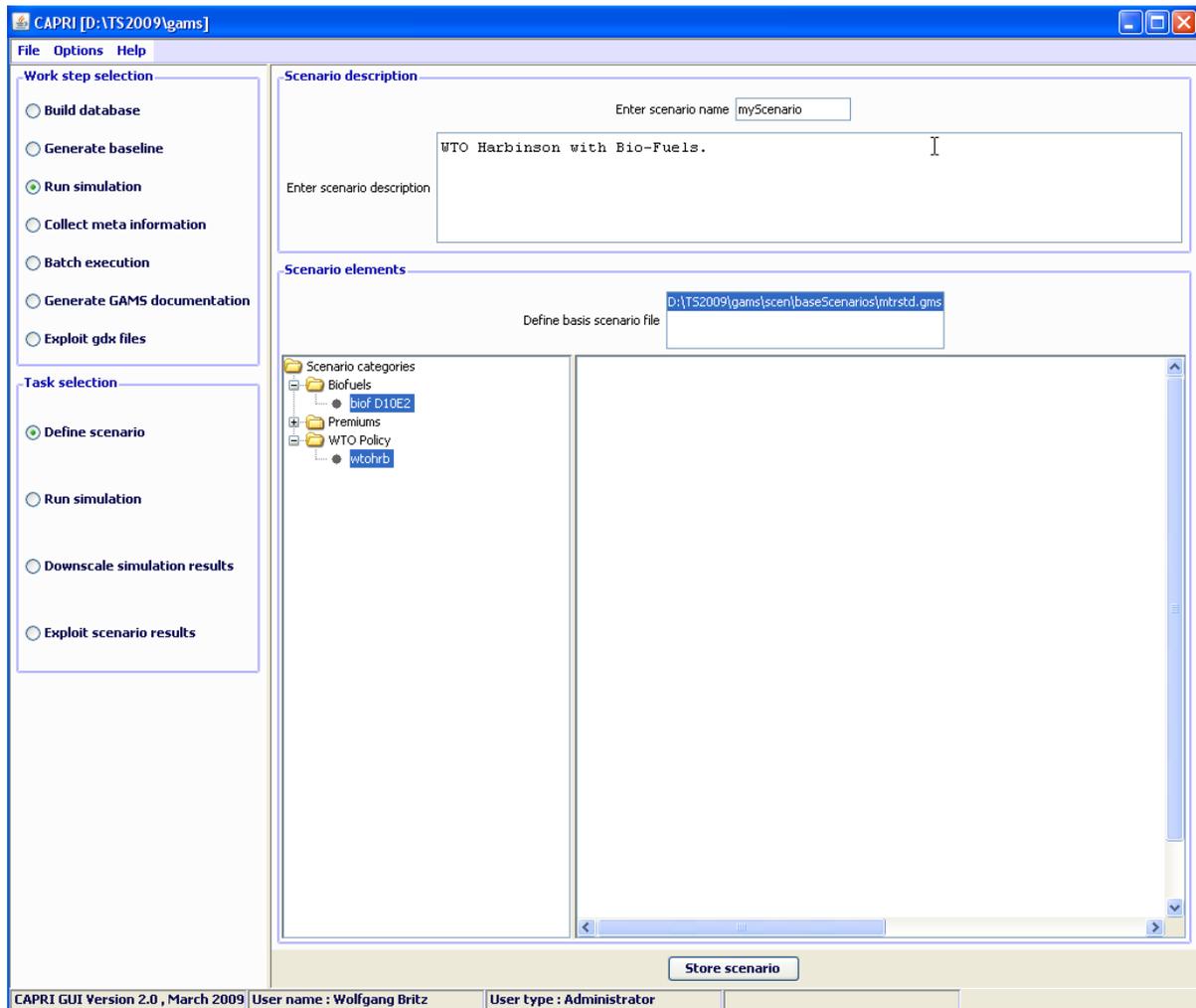
# The task run simulation

Graph: the task panel for “Run Simulation”

**Task selection**

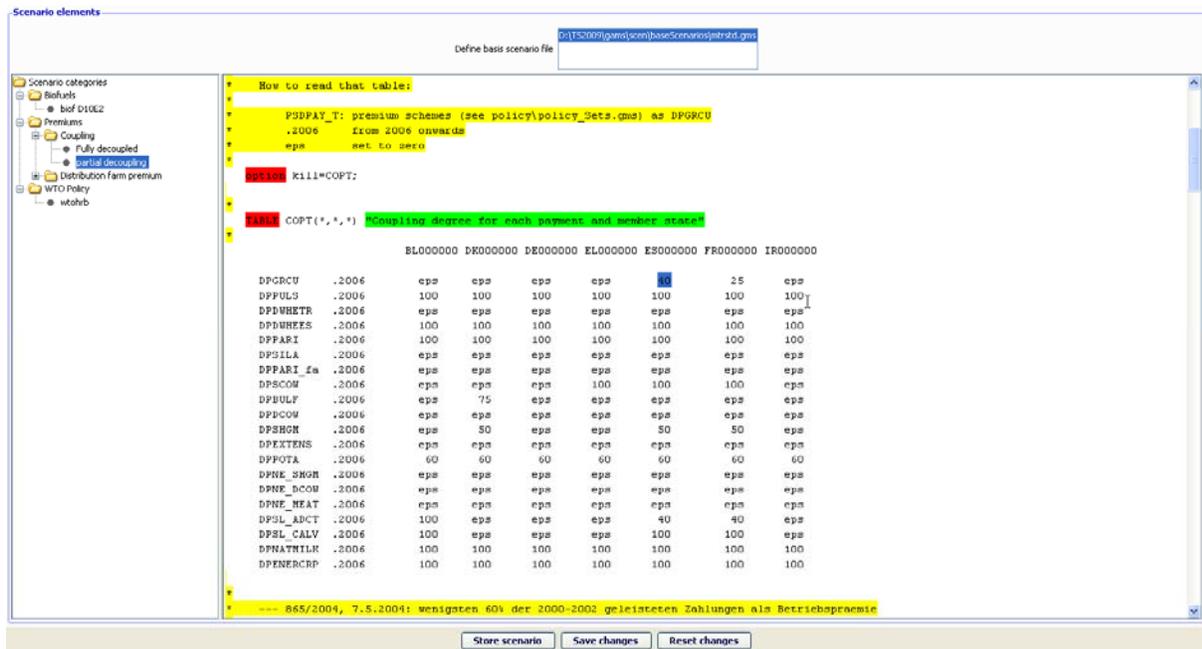
- Define scenario
- Run simulation
- Downscale simulation results
- Exploit scenario results

## Define scenario task



Choosing the task adds the panel with GUI elements shown above. The panel consists of two major panes:

1. A top pane where the user can enter a name for his new scenario, and a description text.
2. A bottom pane where the user can define the base scenario to start with (currently in the trunk “MTDSTD.gms”) and the snippet to add. The available snippets and their structure is shown on the left hand side in an expandable tree which shows the sub-directories found under “gams\scen”, with the exclusion of a sub-directory called “baseScenarios” and the “.svn” directories. Empty directories are not shown. The user may select any number of snippets, even several from the same sub-directory. Double-clicking on one of the snippets shows the content of the file on the right hand side, so that the user can inspect the code as seen below in more detail. GAMS keywords are shown in red, comments in yellow and strings in green. He can also edit the file – changes are shown in blue. Once changes had been saved, the tree shows a (user modified) behind the category. The user can also remove the changes from snippets.



Storing the scenario then generates a file as shown below, user name, the reference to CAPMOD.GMS and the date and time are automatically added by the GUI. The files will be added to the files stored in “gams\pol\_input”.

## Run simulation task

At the core of CAPRI stands its simulation engine, which iteratively links different types of economic models: aggregate programming models at regional or farm type level, with an explicit representation of agricultural production technology, aggregated versions of these models at Member States model linked together to derive market clearing prices for young animals, and finally, a global spatial multi-commodity model for main agricultural products and selected secondary processed products.

Differences in results between simulations may be rooted in three different blocks:

1. Differences in the in-going base year data and baseline. CAPRI allows several base years and calibration points to co-exist, and user may choose the base and baseline year.
2. Difference in what economic models are linked together and in the regionalisation level as the user may switch the market modules on or off, may run the model at Member State NUTS II and farm type level or in comparative static or recursive dynamic mode.
3. And finally, the most common, differences in the exogenous assumptions including the policy definition.

## Graph: The interface for the task run simulation

Task properties for : run simulation

Base year	2002
Simulation year	2013
Member States	BL (Belgium & Luxembourg) DK (Denmark) DE (Germany)
Regional Break down	Member States
Number of iterations	15
Scenario file name	01GWPBUR.GMS
<input type="checkbox"/> Global market model	<input type="checkbox"/> Young animal market model
<input checked="" type="checkbox"/> Aggregates for activities and commodities	<input checked="" type="checkbox"/> Environmental indicators
<input type="checkbox"/> Life-cycle assessment for energy	<input type="checkbox"/> Multi-functionality indicators

The following discussed the settings:

- **Base year:** determines the three year average underlying the regional (see *Build regional data base*) and global data base (see *Build regional data base*) and the trends (see *Generate trend projection*).
- **Simulation year:** the year for which results are generated and trends are loaded.
- **Member States:** if the global market model is switched off, the user may run a simulation for selected Member States, only.
- **Regional break down:** the level of regional dis-aggregation in the supply part. It is not longer recommended to user the “Member State” level for production runs.
- **Number of iterations:** with market models switched on, CAPRI sequentially calibrates the market models to supply model results which are solved at prices from the market models. Usually, fifteen iterations are recommended.
- **Scenario file name:** the GAMS file which comprises the settings for policy and further exogenous variables for a simulation. The files are stored in “gams\pol\_input” and must be valid GAMS code. Use a text editor as e.g. the GAMS GUI to manipulate the files and generate new ones.
- **Global market model:** Switch the spatial global market model for agricultural products on and off. If switched off, outputs prices will be fixed to the baseline results. If switched on, the supply model will work with prices provided by the global market model, and the global market model will be iteratively calibrated to the results of the supply models aggregated to Member State level.

- *Young animal market model*: switch endogenous prices and market clearing for young animals on and off.
- The last four settings relate to post model reporting. Currently, the reporting part is still embedded in the GAMS code and cannot be started separately. That means that the full runs needs to be repeated if results from one of the reports are later needed.

# The task: Exploit scenario results

The task allows loading simultaneously results from different scenarios into the exploitation tools.

## Graph: The interface in exploit scenario result mode

Task properties for : exploit scenario results

Base year	2002
Simulation year	2013
Member States	BL (Belgium & Luxembourg) DK (Denmark) DE (Germany)
Regional Break down	Member States
Scenario 1	
Scenario 2	
Scenario 3	
Scenario 4	
Scenario 5	

[show results](#) [show meta data](#)

## The task: Collect meta information

The task scans the GDX files from the different tasks and collects Meta information (user, date of execution etc.) from there and shows it in a table. Besides the result directory stored in the settings, the task performs a “passive crawl” such that it will collect information on all files in the result directory which could technically be generated by the tasks. Settings as base year or simulation year in the interface are not taking into account. The processing depends on availability of the META symbol in the GDX file. Using the new GUI with files based on older version of CAPRI could lead to situations where files are not included in the reporting system.

The following information is provided:

- **Member State:** the Member State for which Meta information is provided. Depending on the task, the program may have run for that Member State separately (e.g. CAPREG) or along with other Member State (e.g. CAPMOD)
- **Work step:** the task run as reported in the GDX file.
- **#:** logical order of the work steps. Steps with a higher order build on results of steps with a lower one. Consequently, results from steps from steps with a higher order should be younger than those from steps with a lower order.
- **Item:** the different types of Meta data provided.
- **Content:** actual Meta data for the item.
- **File name and SVN status:** the GDX file from which the meta information is loaded, plus information about its status in the SVN versioning system. “Modified” means that the file in the local copy was modified since the last update and was not committed afterward. “Out of date” means that a newer version is available on the server. “Conflicting” means that an updated version of the file is available to the local copy, but that the file was modified. “Not under version control” means that the file is not handled by the version system.

It allows checking the internal consistency, i.e. if results entering certain work steps are outdated. In that case, the line with the date stamp is shown in red.

The scroll down boxes above the table can be used to select in the table. The first entry (\*.\*) selects all items. Clicking on the table columns allows sorting.

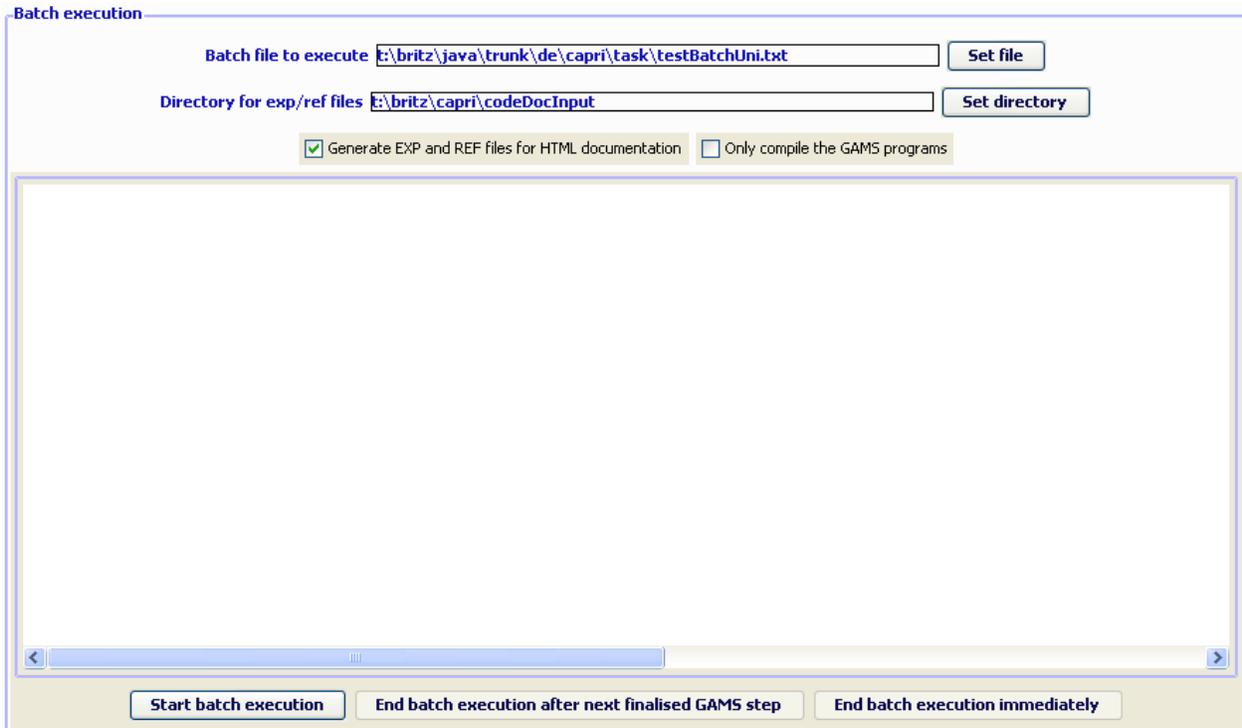
## Graph: Table with meta information

Meta data information

Member state	WorkStep	#	Item	Content	File name and SWN status
AL000000	PREPARE NATIONAL DATABASE	0	TITLE OF DATA SET	Prepare national database COCO1	coco_output_trp.gdx : Normal
AL000000	BUILD REGIONAL DATABASE	1	DATE OF VERSION	2008-05-28 18:09:34	res_02.gdx : Modified
AL000000	BASLINE CALIBRATION	4	TITLE OF DATA SET	Baseline calibration MTRSTD	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	WORKSTEP	Generate baseline	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	KEY	MTRSTD	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	NAME OF PROCESSOR ORGANISATION	Wolfgang Britz	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	TEMPORAL COVERAGE	2013	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	DATE OF VERSION	2008-09-26 15:53:45	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	BASEYEAR	2002	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	SIMYEAR	2013	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	MODEL_SWITCHES	REC_DYN OFF MARKET_M OFF YANI_M OFF	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	MEMBER_STATES	BL DK DE EL ES FR IR IT NL AT PT FI SE UK CZ EE HU LT LV PL SI SK RO BG CY MT NO AL MK CS BA HR MO KO	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	LANGUAGE WITHIN THE DATA SET	ENGLISH	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	NAME OF EXCHANGE FORMAT	GDX	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	NAME OF OWNER ORGANISATION	CAPRI network	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	NAME OF ORIGINATOR ORGANISATION	CAPRI network	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	DESCRIPTION OF PROCESS STEP	Generate baseline, Baseline calibration	res_2_0213MTRSTD.gdx : Modified
AL000000	BASLINE CALIBRATION	4	GEOGRAPHIC COVERAGE BY NAME	BL DK DE EL ES FR IR IT NL AT PT FI SE UK CZ EE HU LT LV PL SI SK RO BG CY MT NO AL MK CS BA HR MO KO	res_2_0213MTRSTD.gdx : Modified
AL000000	GENERATE EXPOST RESULTS	2	TITLE OF DATA SET	Generate expost results EXPOST	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	WORKSTEP	Generate baseline	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	KEY	EXPOST	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	NAME OF PROCESSOR ORGANISATION	Alexander Godt	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	TEMPORAL COVERAGE	2002	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	DATE OF VERSION	2008-09-01 11:25:55	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	BASEYEAR	2002	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	SIMYEAR	2013	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	MODEL_SWITCHES	REC_DYN ON MARKET_M ON YANI_M ON	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	MEMBER_STATES	BL DK DE EL ES FR IR IT NL AT PT FI SE UK CZ EE HU LT LV PL SI SK RO BG CY MT NO AL MK CS BA HR MO KO	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	LANGUAGE WITHIN THE DATA SET	ENGLISH	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	NAME OF EXCHANGE FORMAT	GDX	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	NAME OF OWNER ORGANISATION	CAPRI network	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	NAME OF ORIGINATOR ORGANISATION	CAPRI network	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	DESCRIPTION OF PROCESS STEP	Generate baseline, Generate expost results	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE EXPOST RESULTS	2	GEOGRAPHIC COVERAGE BY NAME	BL DK DE EL ES FR IR IT NL AT PT FI SE UK CZ EE HU LT LV PL SI SK RO BG CY MT NO AL MK CS BA HR MO KO	res_2_0202mtrstd.gdx : Normal
AL000000	GENERATE TREND PROJECTION	3	TITLE OF DATA SET	Generate trend projection CAPTRD	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	DATE OF VERSION	2008-09-24 22:53:35	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	TEMPORAL COVERAGE	1984 - 2030	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	LANGUAGE WITHIN THE DATA SET	ENGLISH	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	NAME OF EXCHANGE FORMAT	GDX	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	GEOGRAPHIC COVERAGE BY NAME	BL DK DE EL ES FR IR IT NL AT PT FI SE UK CZ EE HU LT LV PL SI SK RO BG CY MT NO AL MK CS BA HR MO KO	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	NAME OF ORIGINATOR ORGANISATION	CAPRI network	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	NAME OF OWNER ORGANISATION	CAPRI network	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	NAME OF PROCESSOR ORGANISATION	Wolfgang Britz	results_0213.gdx : Modified
AL000000	GENERATE TREND PROJECTION	3	DESCRIPTION OF PROCESS STEP	Generate baseline, Generate trend projection	results_0213.gdx : Modified

# Batch execution

Graph: Batch execution panel



The batch execution allows starting a file defining settings and tasks from the different CAPRI work steps and executing them without user intervention. Once started, the batch processor may be stopped so that the currently running GAMS programs ends on its own (“end batch execution after next finalised GAMS step”) or by sending a “CTRL-C” to the GAMS program. It will continue to run until the GAMS processor notices the CTRL-C – which may take a while – and then end with an error code. However, the GAMS processor will run some finalisation tasks as removing temporary files and directories.

The batch execution facility is a tool which:

- Allows executing many different CAPRI task after each other without requiring user input.
- Reports the settings used, any errors and GAMS result codes in a HTML page from which they may queried at a later time.
- Ensures that each new run generates its own listing file, which can be opened from the HTML page.
- Allows storing the output of the different runs in a separate directory, while reading input from unchanged result directories.

The purpose of the batch execution facility is therefore at least twofold. At the one hand, it allows to set up test suits for the CAPRI GAMS code such as checking for compilation without errors for all tasks and different settings such as with and without market parts etc.. Secondly, production runs of e.g. different scenarios can be started automatically. It is planned to add timer facilities to the batch execution so that the GUI will start a suite of runs at a pre-scheduled time. Along with the planned functionalities to compare in a more or less automated way differences in results between versions, the batch facility is one important step towards quality control.

For details on the batch execution facility see the technical document “Batch execution of CAPRI tasks” to be found on the Capri web page under technical documents.

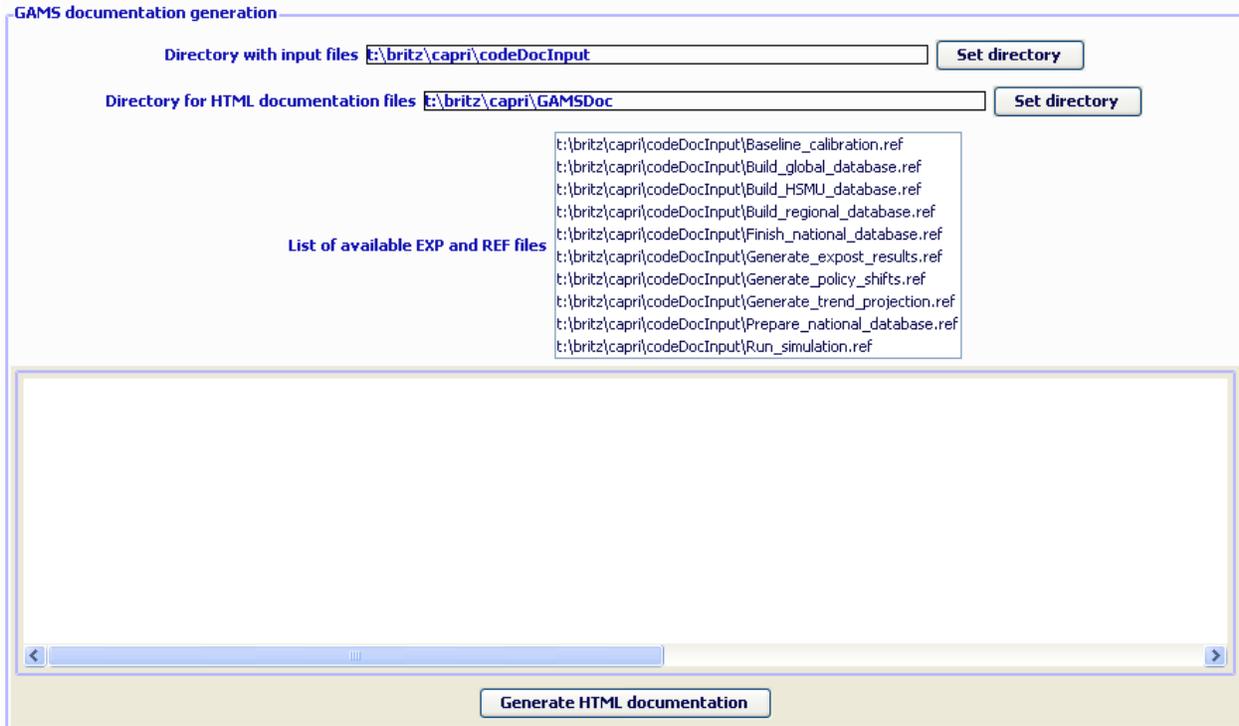
If the suite of tasks comprises “execute” statements, those can be downgraded to “compile” with “Only compile GAMS programs” check box.

The check box “Generate EXP/REF files for HTML documentation” adds settings to the GAMS calls which generate two specific reference files by the GAMS compiler which comprise information of files and symbols used by GAMS. For details on the code documentation facility see the technical document “Javadoc like technical documentation for CAPRI” to be found on the Capri web page under technical documents. The “directory for exp/ref files” defines where those files will be stored.

The batch language allows definition of a timer, i.e. to start the execution at a specified time.

# Task Generate GAMS documentation

## Graph: Panel to steer GAMS documentation generation



The GUI comprises a tool to generate for each GAMS file and each symbol used HTML pages which are interlinked. For details on the code documentation facility see the technical document “Javadoc like technical documentation for CAPRI” to be found on the Capri web page under technical documents.

The controls on top allow the user:

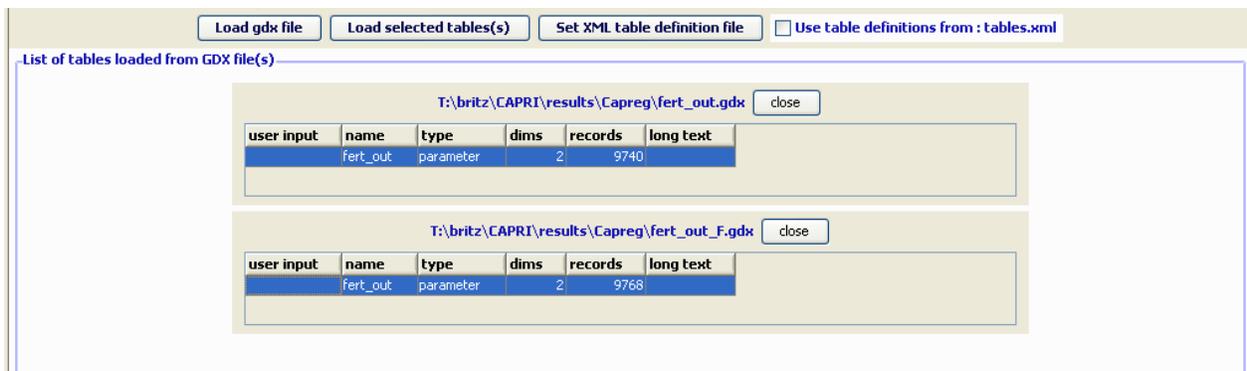
- To define in which directory the “EXP”, “REF” and “GDX” files are stored which serve as input into the documentation generator.
- To chose the directory where the HTML files will be generated.
- To select the tasks covered by the documentation generator.

## Exploitation of.gdx-files

GDX-files are generated by GAMS and typically serve either an exchange format between different GAMS applications, or for exploitation purposes as the GAMS-IDE comprises a view for GDX-files. Further tools for GDX-files are available from GAMS company and are described in different documents. In opposite to listings generated by GAMS programs, the GDX files store the data in full numerical precision in an internal format.

The new CAPRI version passes information from one task to the next with the help of GDX files, so generates CoCo a.gdx files with the time series at national level, which is read by CAPREG. And the regional time series generated by CAPREG are inputted by the trend projection tool CAPTRD. These.gdx files are accessed when the different tasks of “Data base exploitation” are chosen. The user has on top the possibility to load one or several tables from one or several freely chosen.gdx files.

### Graph: Panel to GDX file exploitation



When the task “exploit.gdx files” is selected by pressing the related button, four buttons are shown in the task panel. The first one, labelled “load.gdx files” will open a file selection menu when pressed. When the ok button of the dialogue is operated, the content of the.gdx file is partially loaded, and a table is added to the right upper window of the application showing the parameters and sets comprised in the.gdx files, along with their number of dimensions and records. When the close button next to the table is pressed, the table is deleted. Pressing the “load.gdx file” again will add more tables.

One parameter from each table may be selected (pressing the “ctrl” key when clicking with the mouse de-selects). If several parameters from one file need to be loaded, the user may open the same file several time.

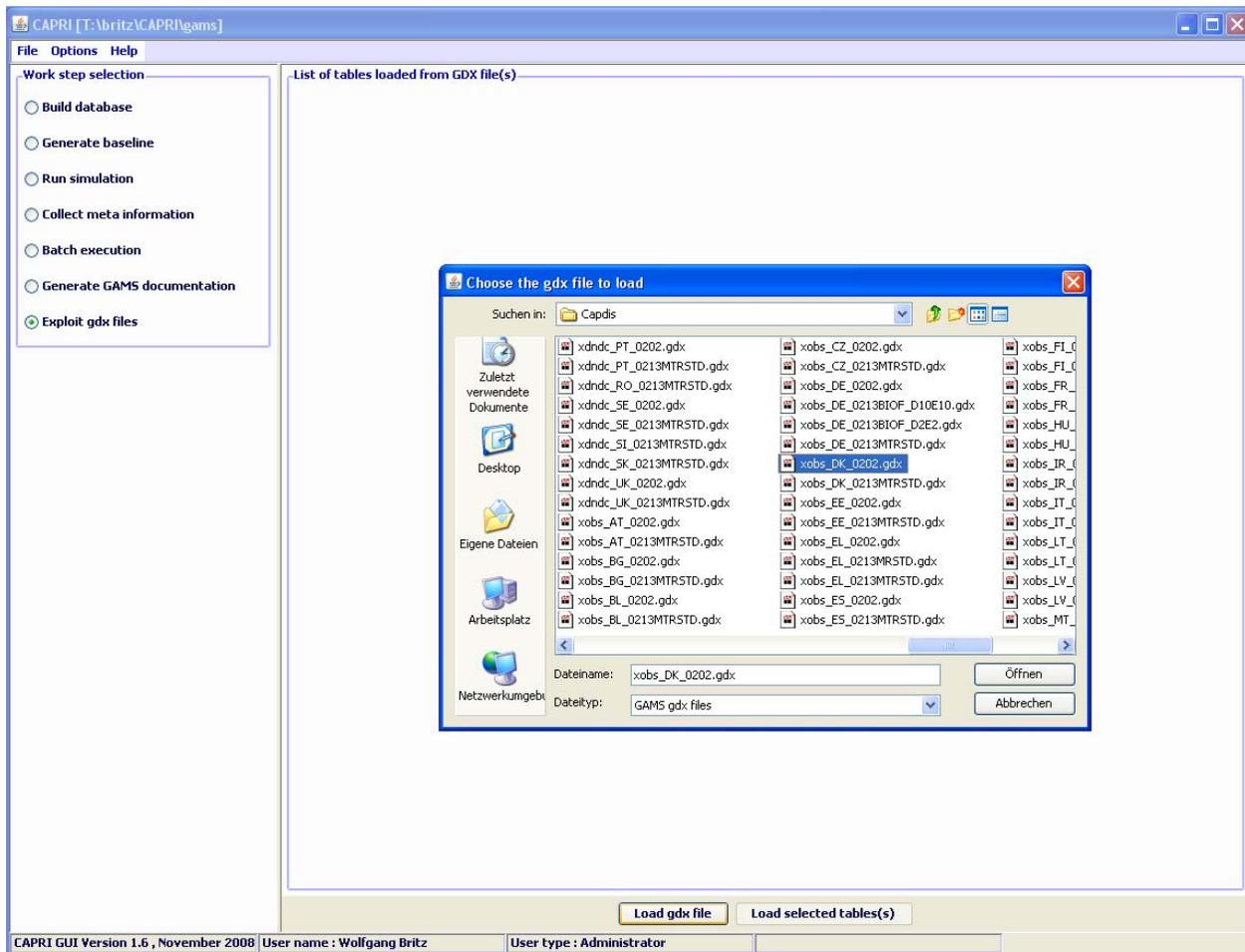
The content of the different parameters is merged together, and the parameters themselves span an additional data dimension. If the user does not provide input in the first column of the tables labelled “user

input”, the program will generate automatically names. The data loaded are shown in the table tool described above.

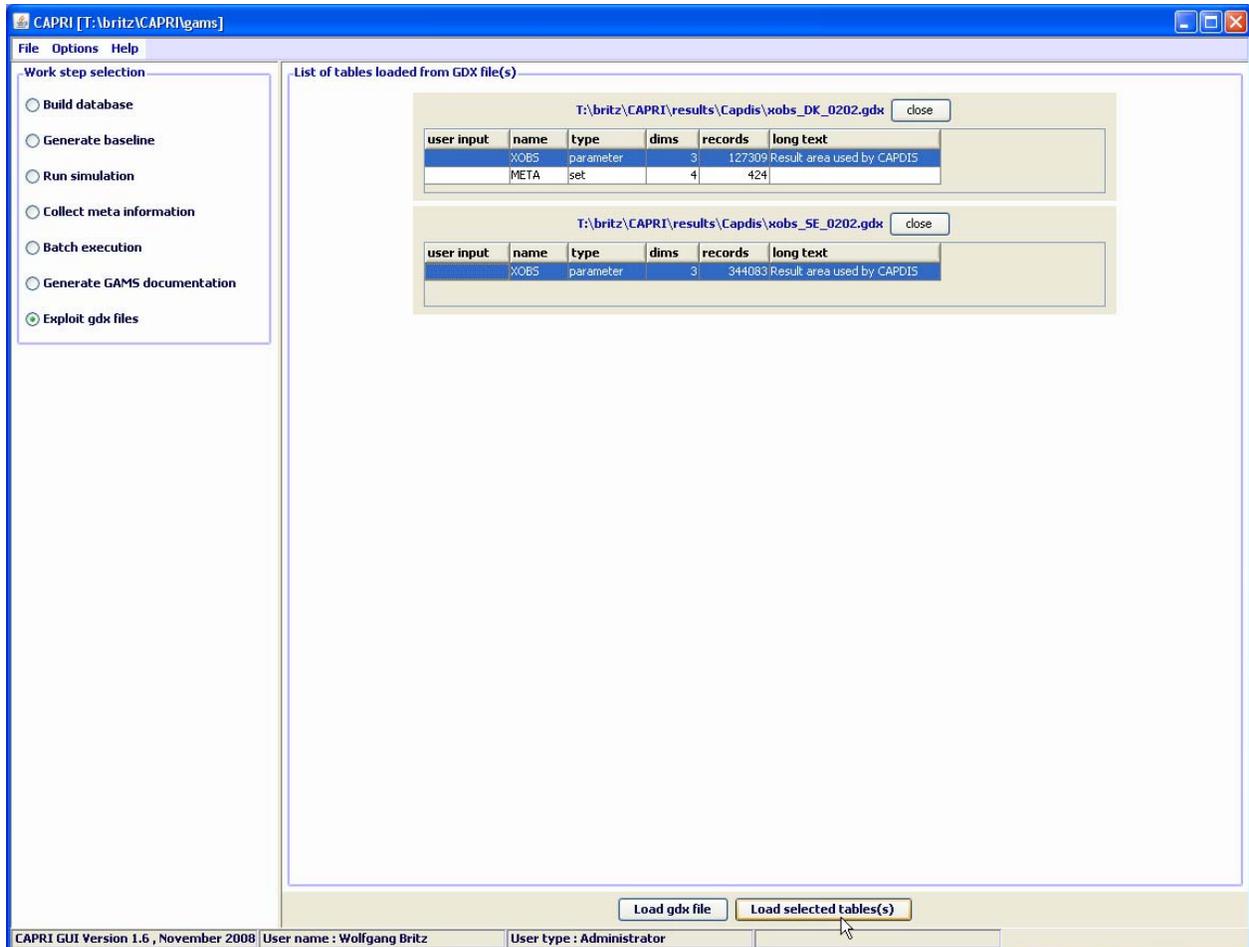
The user can use view definitions stored in a XML file to the tables by pressing the enabling the ”Use table definitions from ...” tick box, and may use the “Set XML table definition file” button to change the file to use.

## An example: loading data from the spatial downscaling

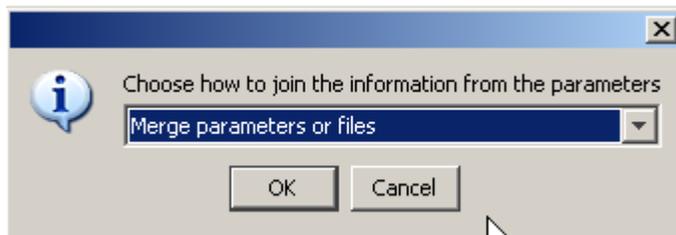
The option described here is introduced for completeness. The names of the file generated by the disaggregation programs start with “XOBS\_” followed with the two character code of the Member state, then an underscore followed by the base year and the simulation year, and if applicable, the code for the simulation which is identical to the name of the GAMS files used from “pol\_input” which was used to run the scenario.



If one wishes to see the information for several Member States simultaneously, one may click on “Load.gdx file” again and add additional files. Afterwards, the lines with the parameters “XOBS” should be selected.



Pressing load selected tables will open the following dialogue, choose



As otherwise, the program will introduce a new dimension for the data loaded from the different files, and you will not be able to see the information for “DK” and “SE” together in one map. Afterwards, the selected records will be loaded from the files.

Depending on the amounts of records, that may take several seconds. Before turning to the mapping view, only one column should be selected. Press on the button left of “SWHE” in the table headers (“Open

selection dialog for table column”), and select with the mouse one of the codes, then press o.k. The table should now comprise only one column. Afterwards use the drop-down list with the viewing options and choose map as shown below. Choose “HSMU.” to select the geometry for the HSMUs. The program will now load the geometry for the HSMU and draw the map which takes several seconds.

## Interaction with GAMS

The interaction with GAMS consists of three parts:

- Generating GAMS code based on user input
- Starting GAMS
- Controlling the GAMS run

There are two types of input files generated based on user input. The first one are so-called scenario files and define the exogenous drivers for a CAPRI run as population growth, macro-economic environment or policy definitions. Here, the final aim is to integrate the scenario editor from SEAMLESS into the CAPRI user interface. The scenario files are typically stored for longer period on disk, both to provide templates for other scenarios as well as for documentation purposes. The name of the file to load is passed to GAMS either as an argument or stored in a input file with a fixed name. The second types are rather small files with a fixed name which typically comprise the information for which years and regions to run the GAMS program along with a small number of methodological switches. These files are overwritten with each start of the related GAMS code.

GAMS is started as a sub-process in an own thread. The output from GAMS which is typically shown in command processor window is redirected into a pipe and its content read from there and shown in a window on the CAPRI user interface, so that the user can check GAMS execution at run time. The code allows filtering out specific statements generated by GAMS to be shown in the windows title bar to give an indication about program progress.

There are two final control mechanisms. Firstly, the return code by GAMS which indicates if the GAMS program was correctly compiled and then executed. Typical execution time errors are math errors as division by zeros or read/write errors on external files. Secondly, the user can apply different type of exploitation tools to check the logical content of the results.

# Exploitation tools

## The structure of the GAMS generated.gdx files

The exploitation tools load directly the .gdx-files generated by the GAMS processes linked to the tasks described above. The .gdx-files only store non-zero numerical values. The main content of a .gdx file are two types of records. The first type provides a list of all labels used to identify the numerical data in the .gdx file as GAMS does not support numerical indices, but requires character labels. The list does not distinguish for which data dimensions the labels are used. They are hence typically a mix of product, activity, region and further labels. The second type of records belongs to GAMS parameters (scalars, vectors, or multi-dimensional tables). Each non-zero numerical item in each parameter has its own record. Each of these records provides the numerical data in double precision (depending on the parameter type there may be different data stored in one record, as for a variables its upper and lower bound, current level and marginal value etc.), and a vector of indices pointing in the list of codes described above.

## Loading the data from .gdx files

The data matrices generated by the different tasks as described above and stored in .gdx-files are typically rather sparse, so that it seemed appropriate to load the data from the .gdx-file into hash tables for exploitation purposes. That is done in a two step procedure. In the first step, all records from the .gdx file are read and vectors of all found indices are stored. The length of each data dimension is only known when all data records are read, and is equal to the number of unique indices for each dimension. Once all records are read, the final length of these index vectors then defines a linear index room for the multi-dimensional table. In a second step, the records are read again, and the index vectors for each record now allow to define a linear index in the total table. A hash code is derived from that linear index to store the numerical values into a hash table. As the number of items to store in the hash table is known beforehand, a rather simple hash table implementation can be used. If necessary, step one can be run over several parameters which may be hosted in several .gdx files, so that results can from different runs can be merged into one hash table.

As the .gdx-files provide a lists of all labels used in any parameters stored in that .gdx-file, the index vectors allow to build lists of labels linked for each index in a data dimension. There exists an additional storage type in the .gdx-files to retrieve long-texts to the labels as defined in GAMS set definitions. However, one label may occur in different sets with different long texts, and the .gdx-file do not store a possibly user defined relation between a data dimension of a parameter and a specific set, an option termed domain checking in GAMS. In order to link hence long-texts to the labels used for a specific data

dimension, two options are possible. Firstly, the user may interactively at run time re-establish based on its knowledge the link between data dimensions and specific sets, and thus add long-texts to the labels used on that data dimension. Or the relation may be hard coded in the JAVA code.

## Multi-dimensional viewer with pivoting and exporting possibilities

The multi-dimensional table is then loaded in a spreadsheet like viewer with pivot-possibilities. The user may switch between a tabular view of the data, or different type of graphs (line, bar, pie, spider) or maps. Scroll-down boxes allow the user to rotate through data dimension not shown in the view port columns and rows. Several data dimensions may be merged into one view port dimension. The user can use column and rows groups, and may apply selection to columns and rows as well as to columns and column groups. Rows carrying zero values only may be hidden. Rows may be sorted by size of the numerical values in one or several columns. The current table may be loaded into the clipboard. Alternatively, all or a selection of tables may be exported to a external file, in different format (HTML, CSV, tab-separated, GAMS, fixed width tables). There are further possibilities as changing fonts or the number of decimals.

## Pre-defined views

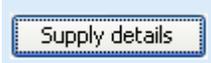
A XML file links pre-defined views to the result content of the tasks. Each view defines selections in the different data dimensions, the view type (table, graph or map) and the pivot, plus some other information.

### Graph: A pre-defined view

	BIOF_D2E2	BIOF_D10E10
Time and date	2008/02/26	2008/02/18
Recursive dynamic mode	OFF	OFF
Expost simulation mode	OFF	OFF
Policy shift mode	OFF	OFF
Baseline calibration mode	OFF	OFF
Market model for agricultural outputs	ON	ON
Market model for young animals	ON	ON
Regional break down (HUTS-level)	2	2
Base year	2002	2002
Number of Member States included	34	34
Number of regional units included	252	252
Number of inner iterations per year	15	15

## View selection

The *currently selected view* is shown as description of the pushbutton in the left corner of the toolbar

under the label *View*.  Pressing the button opens a pop-up menu to *select another view*. The available views will depend on the results you have loaded. The views are logically grouped under

heading, and moving on the heading will show the single views. Some views will be opened as graphics (see chapter ) or maps (see chapter



Selection for rows

## Navigating in the outer dimensions of the viewport

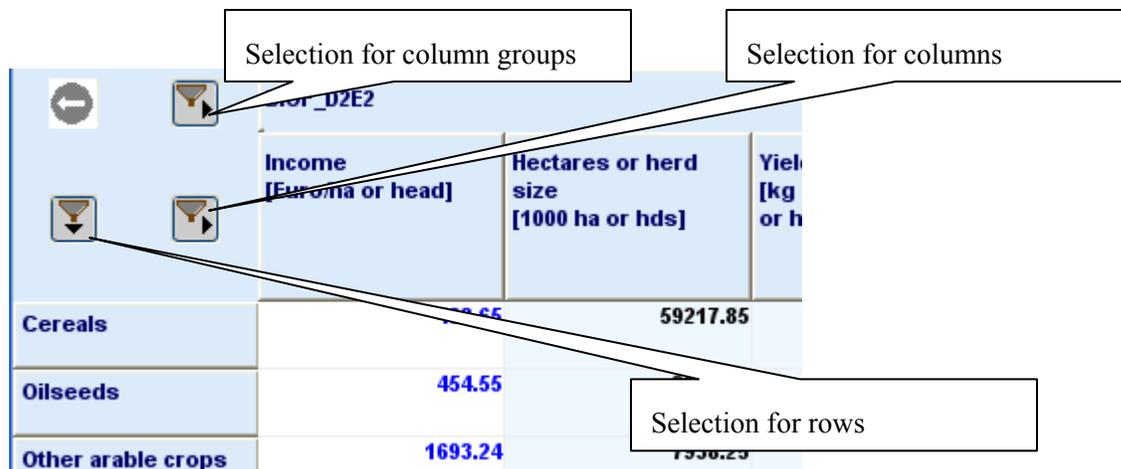
In many views, some data dimensions will not be shown in the columns and rows, but as drop-down boxes in the toolbar. Use the mouse to select in the boxes. You can also use the keyboard to search items by



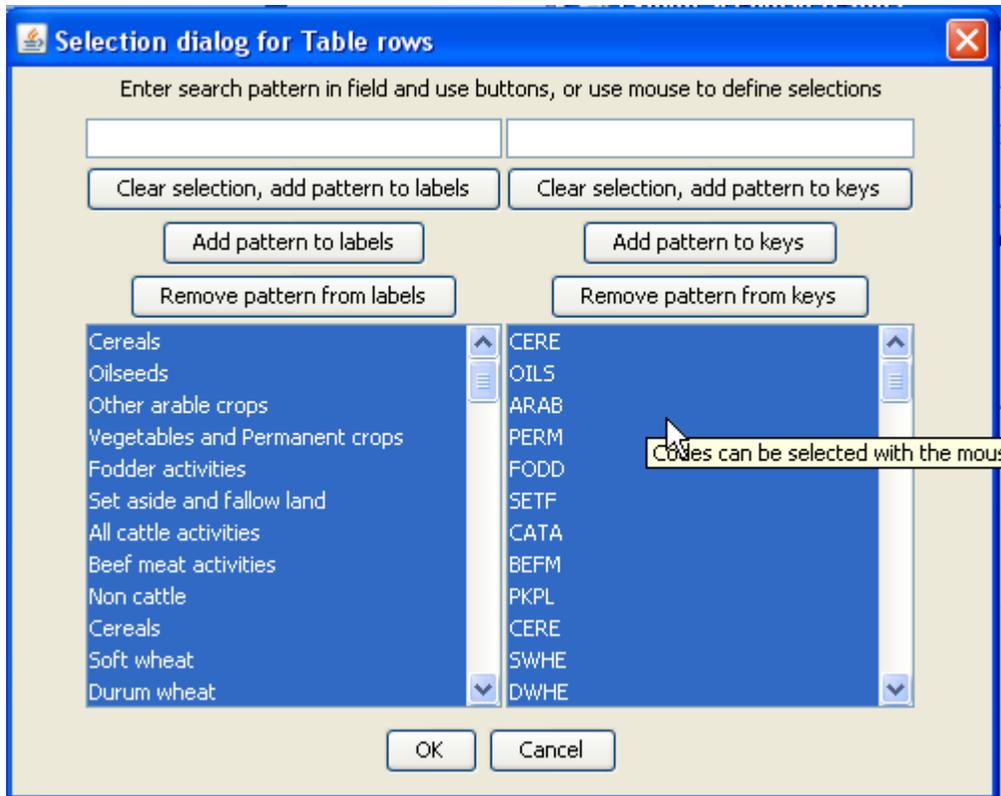
typing. In example for these controls is shown here.

## Column and row selection

Columns and rows can be hidden and included in the current view by using the buttons shown below.

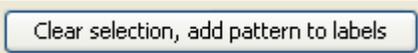
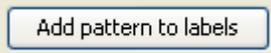


Pressing the button will open a selection dialogue:



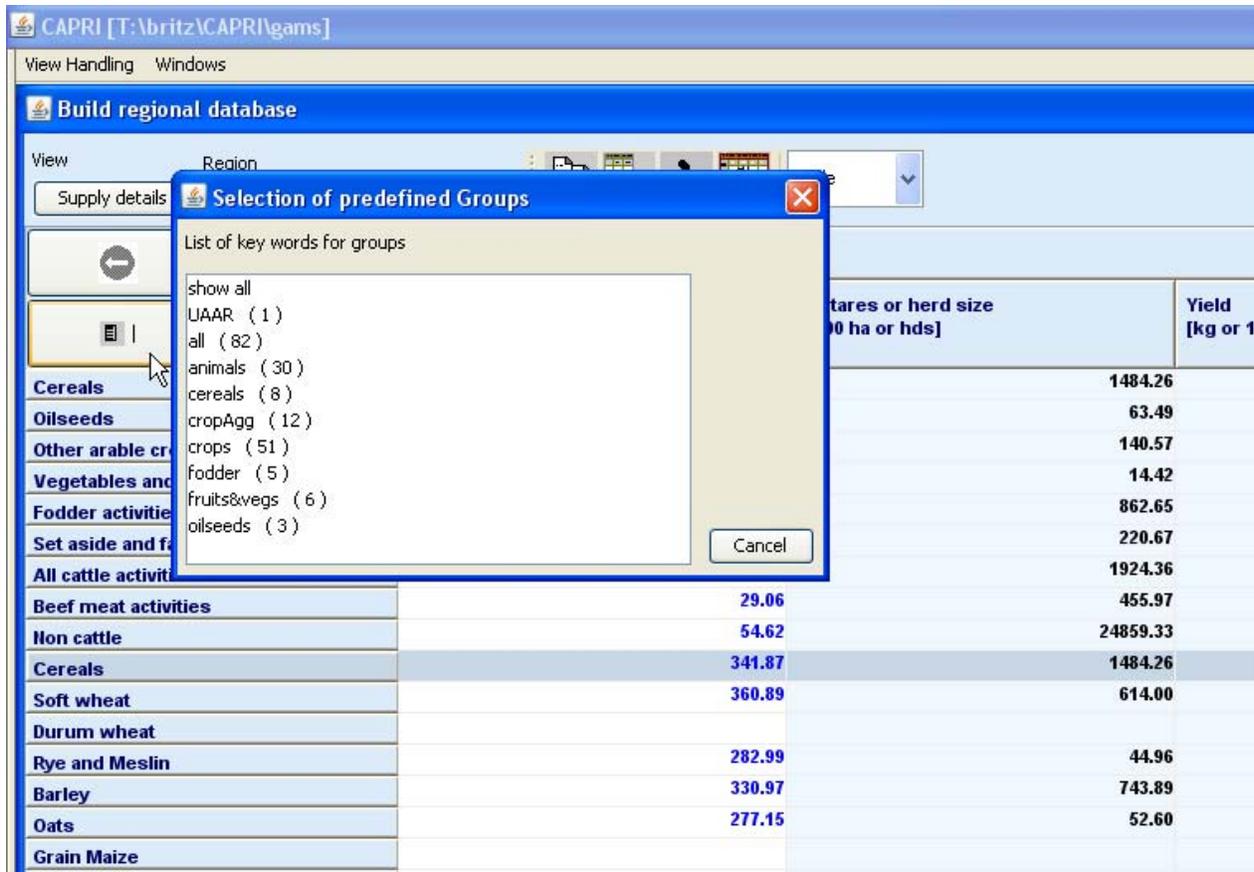
The selections can be done with mouse, following the convention of the operation systems. Additionally, a selection string can be entered in the field above, with the following possibilities:

- “\*” select all
- “C\*” select all items starting with “C”, “C???” will select string starting with C followed by any 3 characters.

The  button will remove any selection and select only those items which match the entered pattern. Pressing  will keep the selection and add the matching items, where  will remove matching items from the selection.

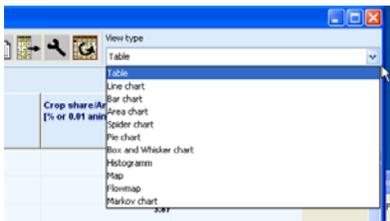
## Predefined selection groups

For some tables, pre-defined selection groups for columns or rows are stored. When the mouse is moved over the selection button and rests there for some time and such groups exists, a dialogue will show as below where the groups can be selected.



## Selection of the view type

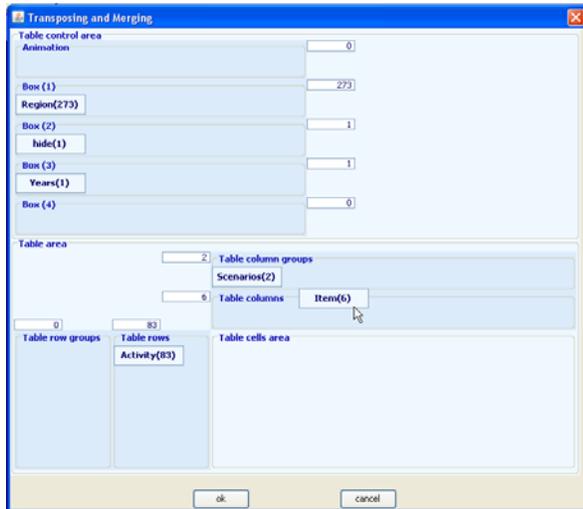
As discussed below, the data can be shown as tables, graphics or maps, to do so use the view selection drop-down box:



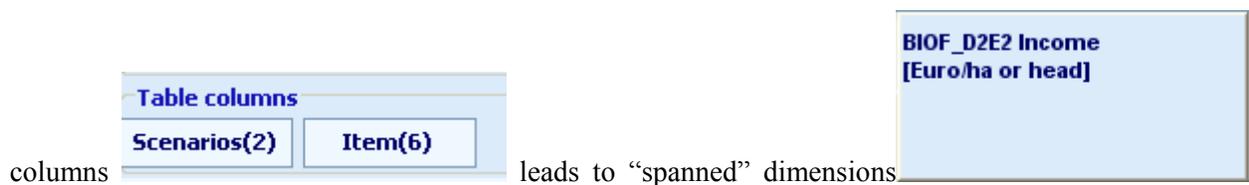
## Manually changing the pivot

Normally, the predefined views will link the data dimension in an appropriate way to columns and rows. However, the user is free to change the pivot, to e.g. generate a cross-sectional series. A dialog opens

when pressing the  button to pivot the currently shown or selected part of the view:



The boxes **Scenarios(2)** show the data dimension and their lengths. They can be dragged to the different viewport dimension as shown in the screen shot above. Assigning several dimensions to the



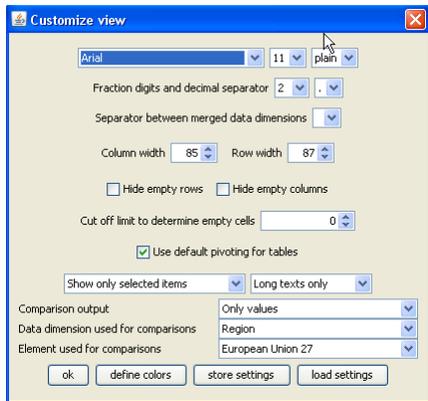
columns leads to “spanned” dimensions

		Cereals	Oilseeds	Other
BIOF_D2E2	Income [Euro/ha or head]	438.65	454.55	
		-17.73%	-20.01%	
	Hectares or herd size [1000 ha or hds]	59217.85	8336.27	
		-2.06%	-13.24%	
	Yield [kg or 1/1000 head/ha or head]	4958.89	7271.59	
		0.85%	6.37%	
BIOF_D18E1	Income [Euro/ha or head]	533.21	648.48	
		0.00%	0.00%	
	Hectares or herd size [1000 ha or hds]	68473.37	9687.46	
		0.00%	0.00%	
	Yield [kg or 1/1000 head/ha or head]	5142.97	2421.83	
		0.00%	0.00%	

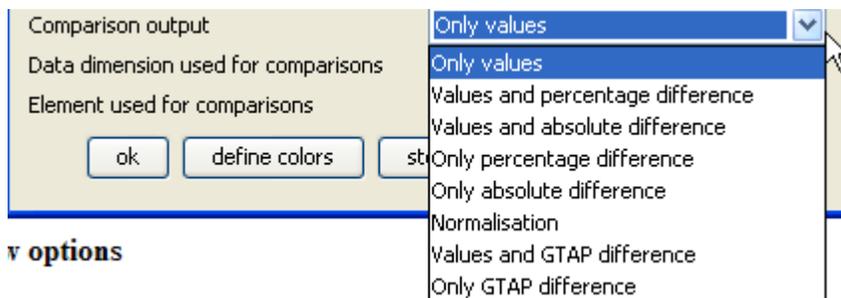
Alternatively, columns and rows can have row block: . In combination with the selections for columns and rows, and column and row blocks, the view can be adjusted to the need of the user, e.g. to export the data in a specific ordering to external file.

## Changing view options

A dialog opens when pressing the  button to change various options of the current view:



- *Fonts*: set font family, size and style - affects tabular views
- *Number formatting*: chose number of digits and define the decimal separator. The tool supports rounding numbers before the decimal point by allowing for negative fraction digits. Choosing e.g. -1 will round all numbers to tens. . The numbers shown in graphics or tables are based after the rounding is applied.
- *Hide empty rows and hide empty columns* will suppress in currently seen view any columns and rows which would show only blank cells.
- *Use of short code and/or long texts*
- *Comparison output*: the exploitation tools can add different types of comparison output. They also affect what is shown in maps and graphics.



#### v options

“Normalisation” means that the value is divided by the comparison points, allowing e.g. also to calculate shares. The “GTAP” difference is a compromise between a percentage and an absolute difference: it multiplies the difference in the logs with the difference (thanks to Rob McDougall from the GTAP team in Purdue for the proposal).

In tables, the “and” options will show two lines in each data cell, one with the observations, and

BIOF_D2E2				
	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Yield [kg or 1/1000 head/ha or head]	Supply [1000 t or 1000 animals]
<b>Cereals</b>	438.65 -17.73%	59217.85 -2.08%	4958.80 -3.58%	293649.34 -5.58%
<b>Oilseeds</b>	454.55 -30.01%	8335.37 -13.24%	2274.59 -6.17%	18934.54 -18.60%
<b>Other arable crops</b>	1693.24 -4.26%	7938.25 2.27%	32743.16 -2.99%	259923.47 -0.78%

one with the comparison output, as seen below.

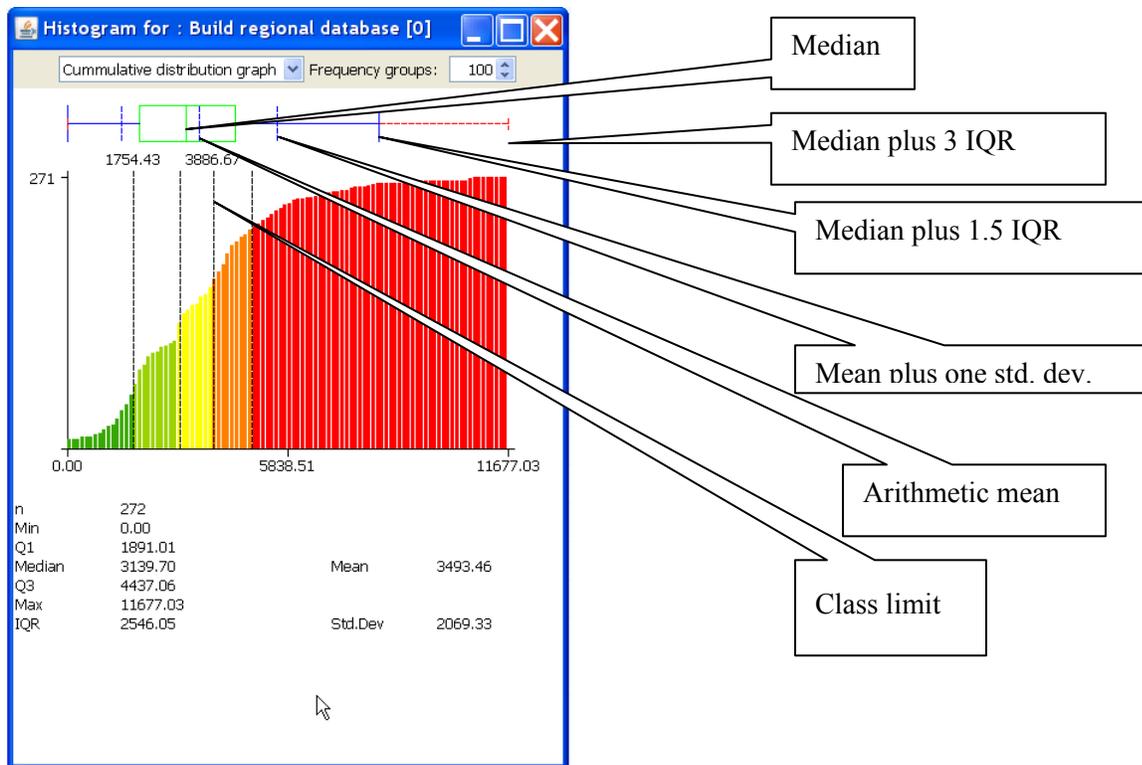
The “Data dimension used for comparisons” offers a drop-down list” with in connection with drop-down list to select the “Element used for comparisons” defined the comparison point.

## Showing a histogram windows

The system offers different ways to retrieve information about the distribution. For maps and tables, the user can show an additional window with a box and whisker diagram, histogram and some descriptive statistics as shown below. The box and whisker diagram is defined as follows: the green box shows the first (Q1) to third quartile (Q3), so that the width of the box is equal to the so-called inner quartile range (IRQ). The blue “whiskers” are defined by Q1 minus 1.5 times IQR and Q3 plus 1.5 times IQR, but bounded by the minimum and maximum of the observations. In many applications, any value falling outside that range is classified as a mild outlier. The red dotted whiskers are at Q1 minus 3 times IQR and Q3 plus 3 times IQR, but bounded by the minimum and maximum of the observations. In many applications, any value falling outside that range is classified as a stronger outlier.

The user can restrict the plotted range as to exclude stronger outliers. If outliers are presents, the red dotted whiskers at the tail with strong outliers are removed.

The blue dotted lines show the mean, and +/- one standard deviation around the mean. For a normal distribution, that would cover around 2/3 of the observations. The black dotted lines in the histogram show the class limits used for the colour model. The bottom reports some descriptive statistics. The technical implementation is set up according to the way maps are drawn: the population consists of all values in the rows and the columns of the table, and thus differs from the outlier control, which treats each columns as a separate set of observations.



The colors are typically used to visualize the distribution in maps, but, as a second option, they can also be applied to the numerical values in tables. Alternatively, histograms and box and whisker diagrams can be drawn via the graphics.

### Working with tables

Tool bar      Controls for column and row selection

	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Yield [kg or 1/1000 head/ha or head]	Supply [1000 t]	Crop share/Animal density [% or 0.01 animals heads/ha]	Production per UAAR [kg/ha]
Cereals			9070.09			29.93
Oilseeds			1485.78			4.90
Other arable crops			1171.71			3.87

### The toolbar



## Tooltips for column and row headers

For predefined tables, tooltips may be stored which give additional information on the columns and rows. They will appear when the mouse is moved over the respective column or row header.

Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Yield [kg or 1/1000 head/ha or head]	Supply [1000 t or 1000 animals]	Crop share [% or 0.01 t]
244.87	4484.26	6228.03	9244.00	
398.57	63.99	2946.76	187.02	
1171.26	140.57	39192.45	5509.33	
30713.87	14.42	285446.28	4116.05	

## Drill-down

Some views comprise a hyper-links to other tables. Numbers with hyperlinks are shown in blue

2515.80
27

, and a tooltip will appear when the mouse is moved over them. Clicking in the cell will jump to the connected table.

## Clipboard export

The content of the currently shown view can be copied to the clipboard by pressing the  button. Tables are placed as tab delimited text in the clipboard, so that they can be pasted into spreadsheets. Graphics and maps are placed as graphics in the clipboard and can be copied into word processing.

## Export to file

A dialog opens when pressing the  button to export the full dataset of the view to file. The action provoked by the button depends on the view type. In tabular view, in opposite to the clipboard export, the export file will scroll through the outer dimensions and will copy all stacked tables after each other into a file. Take the table below as an example. Clipboard export will export the data for Belgium and 1984. File export will export data for *all* regions and for *all* years, if the user does not apply filters in the export dialog. An example is discussed on page 87.

Prepare national database

View: Economic Accounts for Agriculture | Region: Belgium | Years: 1994 | Table: Table

RAW		
	Unit value EAA [Euro/t]	Quantity [1000 t]
<b>Production value</b>		
Cereals		2515.60
Oilseeds		22.00
<b>Inputs</b>		
<b>Premiums</b>		
<b>Production value</b>		
Cereals		2515.60
		4337.40

## Sorting

The rows can be sorted by one or several columns by clicking with the left mouse button in the column headers. A sorting symbol will show sort direction, and its size will show the sorting order.

Cereals	Oilseeds
60473.37	9607.46
0.00%	0.00%
59217.85	8235.27
-2.08%	-13.24%
5142.97	2421.83
0.00%	0.00%

## Numerical filtering based on cell content

Clicking with the right mouse button in one with the columns will open the “filter dialog” which can be used to apply numerical filters to remove rows not matching the filter from the view.

Filter dialog

Define numerical selection filter for table rows

Comparison operator: <>

Comparison value: [ ]

Clear selection and select according to filter

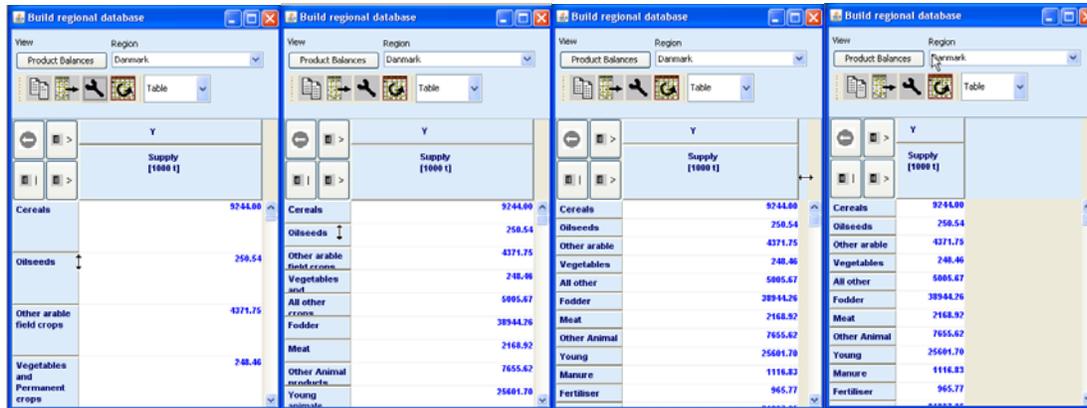
Add result of filter to existing selection

Remove result of filter from existing selection

OK Cancel

## Changing the row height and column width with the mouse

When moving the mouse over the bottom of the first row header cell (Cereals in the example below), the mouse pointer will change as shown below and the cell height can then be dragged with the mouse. The column width can be changed in a similar way by dragging the right border of the column header. Alternatively, the size can set in the “Changing view options” dialogue.

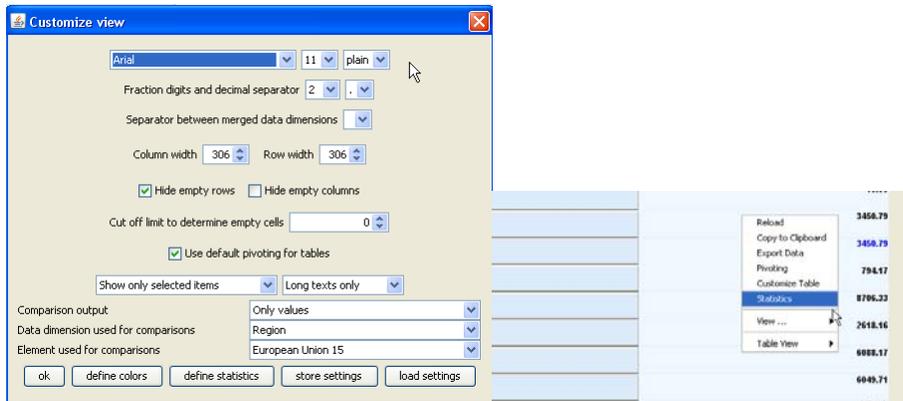


## Adding statistics

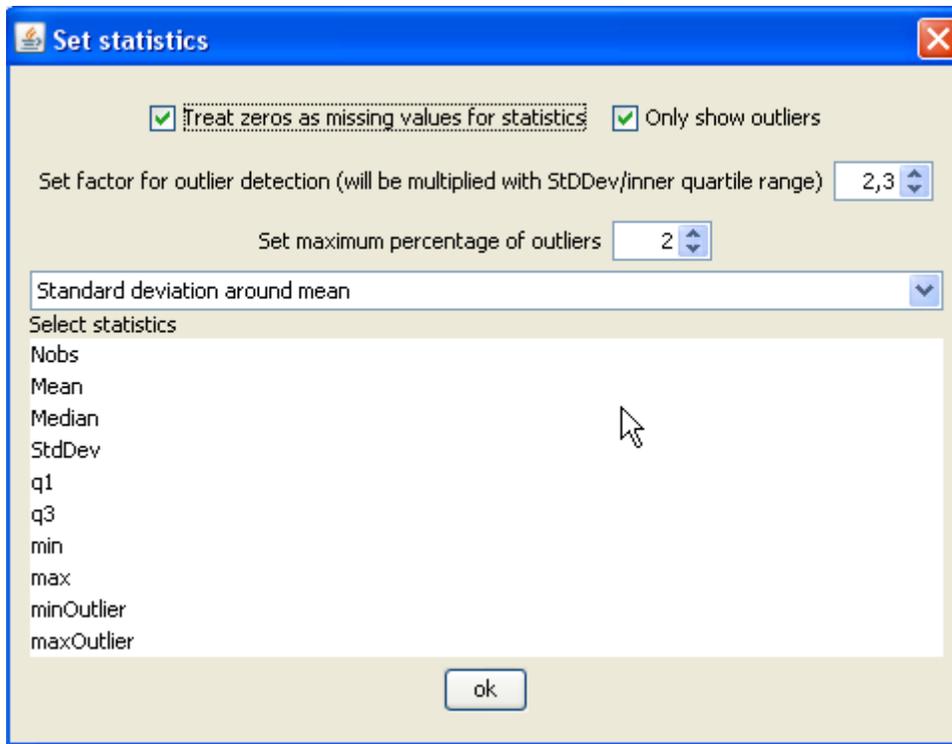
The user may add different statistics as rows to the table as reported in the following table. The observations are assumed to be mapped into the rows of the current views. Zeros can be treated as missing values. The statistics summarize the observation separately for each column.

Statistics	Shortcut
Number of observations	nObs
Arithmetic mean	Mean
Median	Median
Standard Deviation	StdDev
End value in first quartile	q1
First value in fourth quartile	q4
Minimum of the values	min
Maximum of the values	max
Minimum limit for outlier detection as defined from user settings	minOutlier
Maximum limit for outlier detection as defined from user settings	maxOutlier

The related options can be either opened in the “customize dialogue”, using the “define statistics” button, or by a right mouse click in the table to open the popup menu, and choosing



The dialog has the options as shown below, which in parts are dynamically changing depending on the detection algorithm.



Adding the statistics will show them as first rows of the table:

	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Su [10
Hobs	270.00		271.00
Mean	85.89		3001.70
StdDev	108.03		7953.97
Danmark	54.62		24859.33
Germany	64.49		45390.91
Netherlands	22.02		20830.86
France	95.14		42591.81
Spain	58.61		79024.39
Greece	56.83		20299.42
Italy	107.37		24682.36
united Kingdom	52.01		39324.34
Poland	84.33		25153.08
Malta	195.81		154.17
Antwerpen	34.31		1783.06

The perhaps most interesting option is to show besides the statistics only outlier rows in the table, as seen below:

	Income [Euro/ha or head]
Hobs	270.00
Mean	3164.16
StdDev	40975.38
Utrecht	488074.25
Acores	109324.08
Ahvenanmaa/Aaland	-438378.94

## Outlier detection algorithms implemented

The GUI offers currently the following ways to look up possible outliers. For all the methods, the user may additionally define a maximum percentage of observations show in which case only the highest or lower outliers according to the detection algorithm shown will be selected.

### Standard deviation around the mean

The user can define the factor  $\beta$  before the standard deviation. Observations are marked as outliers when their distance to the arithmetic mean exceeds as value defined by the multiplication of the standard deviation  $\sigma$  and a user defined factor  $\beta$ :

$$\neg(\bar{x} + \beta_{\sigma} \cdot \sigma > x_i > \bar{x} - \beta_{\sigma} \cdot \sigma).$$

Large outliers can easily bias the result as they will change both the mean and the standard deviation of the observation sample. Further on, many time series in the CAPRI data base have by definition a lower limit of zero, so that the assumption of normally distributed data set cannot hold. Therefore, other outlier detection methods are also implemented as discussed below. The dialog allows to changing the factor  $\beta$  from its default of 2 which covers for normally distributed data 95% of the values.

### **Standard deviation of values normalized by median**

The values are all divided by the median, and the new series is classified as under 1. The main advantage of that method is the shift to a mid point which is less vulnerable to large outliers in the observations.

### **Standard deviation of trend line error**

A regression is estimated, using the index position in the unsorted values as explanatory values. The resulting errors are then classified according to 1.

### **Median and inner quartile range**

Box-and-whisker charts, which are also supported by the graphics view, are using the median and quartile to visualize the distribution. They are also an easy and robust way to detect possible outliers. First, the so-called “inner quartile range (IQR)” is calculated as the difference between in values between the beginning value of first and ending value of third quartile. The IQR is then the 50% range of the values around the median. The IQR is multiplied with a user defined factor  $\beta$  added to Q3 respectively subtracted from Q1 to define the lower and upper bound for regular values. The factor  $\beta$  default value is 1.5. The quartiles and the median are not affected by outliers at the tails of the distribution, allowing for rather robust way to filter outliers:

$$\neg(Q3 + \beta_{IQR} \cdot IQR > x_i > Q1 - \beta_{IQR} \cdot IQR)$$

### **Conformity based on relation of distances**

Here, the following formulae are used, taken from Last & Kandel (2001):

$$\mu_{i,l} = 2 / \left( 1 + \exp \left( \frac{\beta \cdot n \cdot m \cdot (x_{i+1} - x_i)}{x_{i+m} - x_{i+1}} \right) \right)$$

$$\mu_{i,h} = 2 / \left( 1 + \exp \left( \frac{\beta \cdot n \cdot m \cdot (x_i - x_{i-1})}{x_{i-1} - x_{i-m-1}} \right) \right)$$

They define “conformity” from below and above by comparing the distance from the current value to its neighbour in relation to the average distance for a predefined group size  $m$ . Before the formulae are applied, the values are sorted. In opposite to the outliers based on first and second moment, the method is also able to detect outliers in between “clusters” of values. Inside such a cluster, differences in distances between values are small, so that the relation between the distance to the next neighbour, and the average distance between the neighbour and its  $m$ -th neighbour is around unity. The big advantage of the approach is that it does neither assume a certain functional form for the distribution (as in the case of the mean/standard deviation approach), or a uni-modal distribution is in the case of the IQR method, and it is rather easy to compute. It may be worth to continue with a literature research in the direction of similar outlier detection methods.

The factor  $\beta$  describes how distances between succeeding values are assessed. Outliers are defined when the maximum of the above and below conformity is above a predefined threshold  $\alpha$ .

$$\neg(\max(u_{i,l}, u_{i,h}) > \alpha)$$

Last & Kandel have tested their algorithm for  $\beta=0.001$ ,  $\alpha=0.05$  and  $m=10$ . There seems to be a reach literature on that kind of “neighbourhood distance” based outlier control with different algorithms analyzed in detail. The different parameter can be set by the user interface.

Reference: Last M. & Kandel M. (2001), Automated Detection of Outliers in Real-World Data, Proc. of the Second International Conference on Intelligent Technologies

## ***Working with graphics***

The exploitation tools allow showing the current content of a tabular view as a graphic. Most of the graphic types are based on the JFreeChart library (see <http://www.jfree.org/jfreechart/>).

### **General handling of graphs**

The selection of graphs is based on the drop-down box in the tool bar:

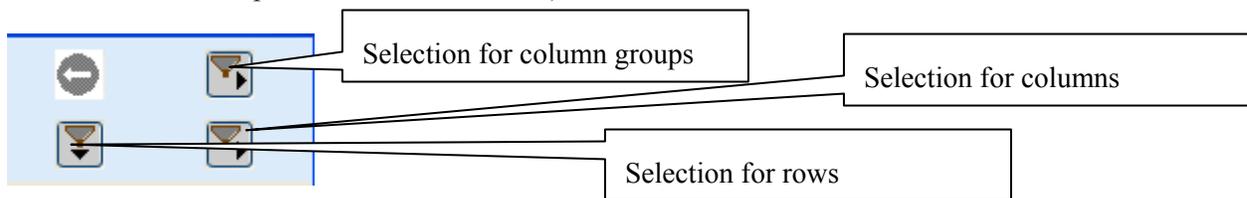


The following *graphic types* are currently supported:

- Bar charts
- Line charts
- Point charts
- Spiderweb charts
- Pie charts
- Box and Whisker charts
- Histograms
- Markov charts

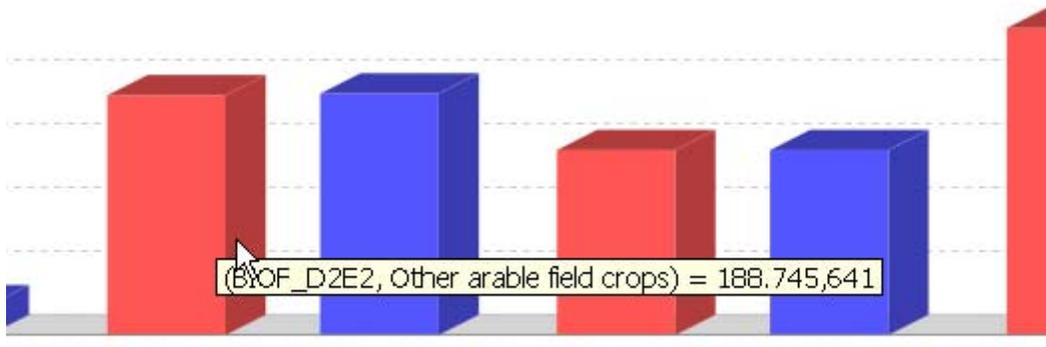
The *selection* of rows and columns shown in the graph can be set in three different ways, for all type of graphics:

- Using the selection dialog (upper left corner of the table, or the buttons next to the graphic type selection drop down box: double click)

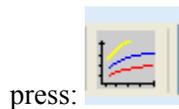


- Using those buttons in graphic mode: single clicks with the left mouse button will scroll down in the list, right mouse single clicks will scroll up.
- Scrolling the table with the scroll bar to a specific position. The column/row in the upper left corner of the table will define the starting point for the graphic.

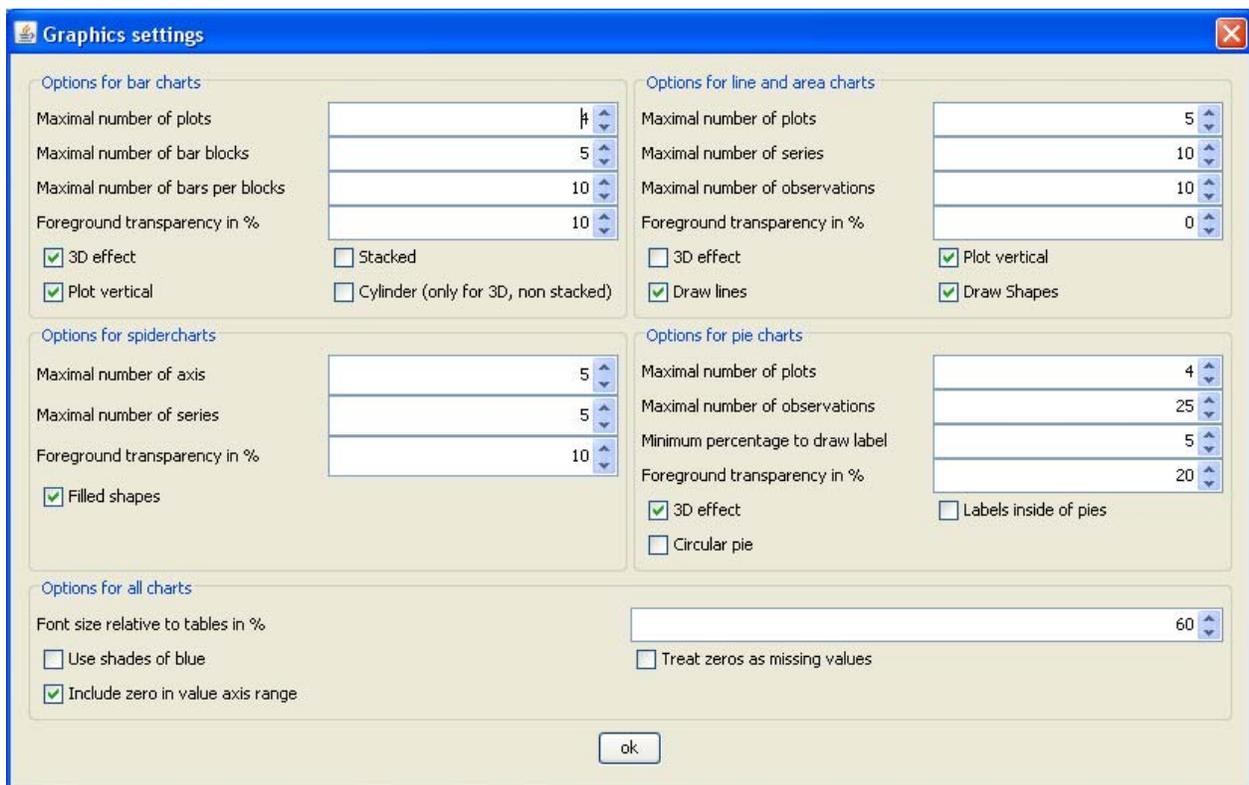
All type of graphics support *tooltips* to query the numerical values underlying the graphic. The tooltips appear when moving the mouse on a graphic element linked to the value as e.g. a bar.



A perhaps unexpected feature is the zooming in and out with the mouse. The graphs support *saving* to disk via a popup menu and *printing*. The popup menu allows also changing for the current graph temporarily certain properties. Some settings which will pertain can be edited opening the graphics option dialogue,



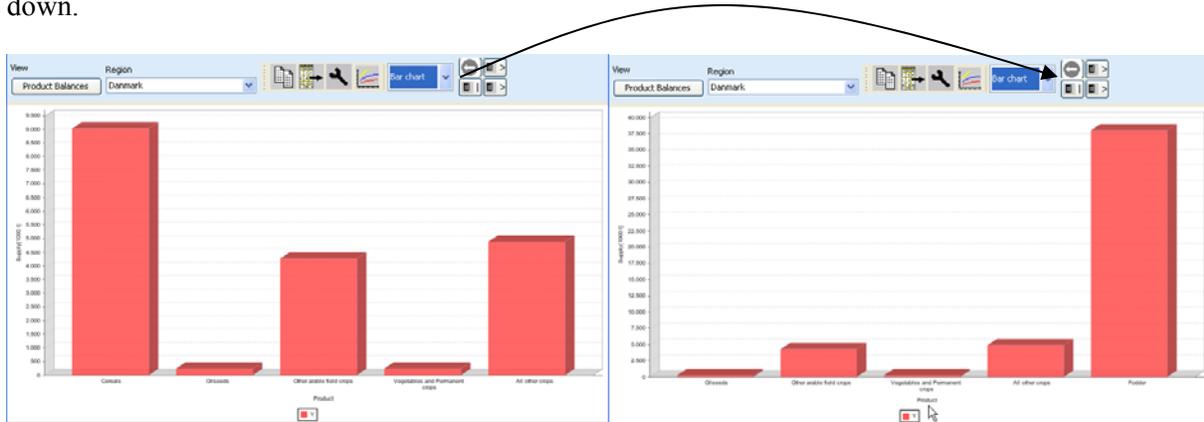
press:



The chart type specific settings are discussed in more detail below. The general options should be self-explanatory, it is best to try them out interactively.

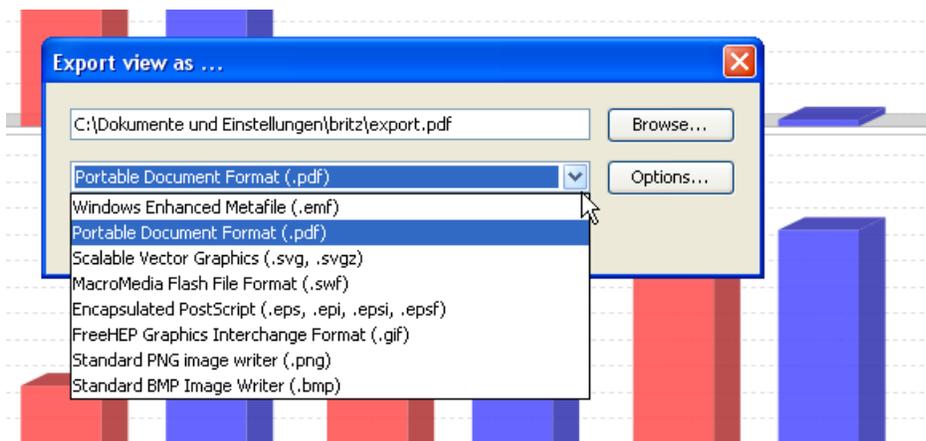
## “Walking” through the data

As not all columns and/or rows will be typically shown in a graph as the maximal numbers of elements shown is restricted (see above), the user has basically two possibilities to change the visible columns or rows. Firstly, columns and rows can be selected by the selection dialogues. Secondly, the user can click with the right or left mouse button on the buttons for table dimensions to mode one row or column up or down.



## Exporting the graphic to file

The graphics can be saved to file in different format by pressing the export  button. The following dialogue will appear which allows the user to define the file, and a range of different file formats. For MS Office users, the “Windows Enhanced Metafile” format is interesting as it allows to later change manually the graphics, e.g. by adding new text.



## Exporting the graphic to clipboard

Alternative, the graphic can be placed into the clipboard where it is stored as a bitmap or as jpeg by

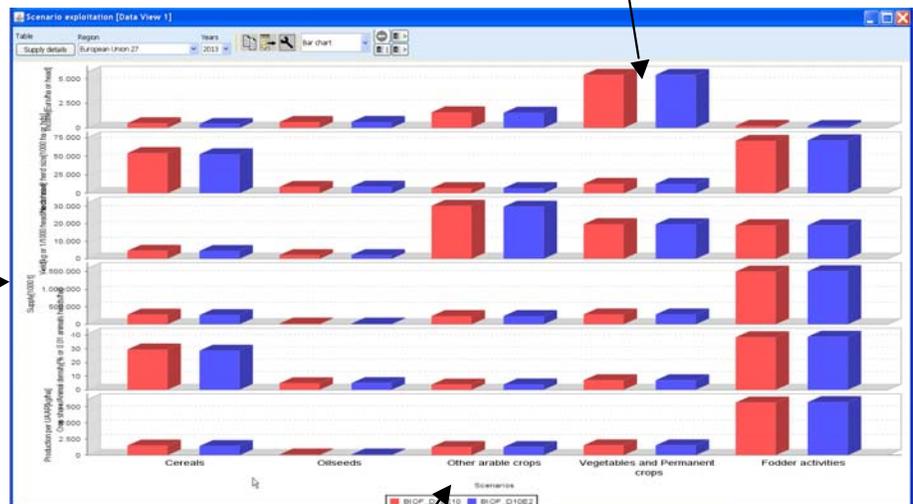
pressing the “copy to clipboard”  button.

## Bar charts

Bar chart treat the columns – typically the table items – as having different units and consequently assign an own plot with a value axis to each of them. The observations are taken from the table rows and define the domain, the horizontal axis. Each column groups are present – typically the scenarios –receives its own colour. An example is given below.



	BI-OP_D10E10					BI-OP_D10E2						
	Income (€1000 ha or head)	Measures or herd size (10000 ha or head)	Yield (kg or 10000 head)	Supply (10000 t)	Crop share/Animal density (% or 0.01 animals/head/ha)	Production per 10000 (kg/ha)	Income (€1000 ha or head)	Measures or herd size (10000 ha or head)	Yield (kg or 10000 head)	Supply (10000 t)	Crop share/Animal density (% or 0.01 animals/head/ha)	Production per 10000 (kg/ha)
Cereals	533.47	60886.55	6154.91	107304.58	32.75	1673.57	653.68	62376.78	6885.55	120656.18	31.75	1584.57
Oilseeds	448.78	8887.27	2388.65	23487.34	6.34	127.85	674.93	16885.66	2378.85	24955.55	5.08	138.78
Other arable crops	4718.55	7885.65	33633.87	161661.71	4.25	1424.31	1744.16	7825.55	22881.81	248788.83	4.31	1488.78
Vegetables and permanent crops	8887.14	16785.21	21678.58	21006.78	7.28	1582.58	1678.16	14211.26	21006.87	210443.28	4.28	1587.83
Fodder activities	278.78	18294.75	21248.89	148188.57	47.43	8887.88	2114.1	22121.87	21121.84	167101.18	43.88	1678.46
Not arable and fallow land	55.34	13384.43	154.71	284.84	7.29	11.24	62.88	13887.38	188.58	2214.46	7.58	12.57
All cattle activities	277.62	14248.88	85.88	1161.88	11.38	44.12	277.88	15082.88	88.85	1653.38	88.88	43.85
Sheep animal activities	4.2	38888.88	348.85	18878.88	16.77	68.16	1.29	28387.62	348.78	18826.84	16.85	72.21



Options for bar charts

Maximal number of plots	#
Maximal number of bar blocks	5
Maximal number of bars per blocks	10
Foreground transparency in %	10

3D effect       Stacked  
 Plot vertical       Cylinder (only for 3D, non stacked)

The user has a number of options for the bar charts. The number of plots refers to the number of columns in the underlying tables, each column will receive an own plot with a matching value axis. The bar blocks refer to the rows, each bar block may comprise several bars taken from the column groups (typically scenarios). As seen above, it is also possible to generate stacked bars from the column groups, or to generate cylinders instead of cubes.

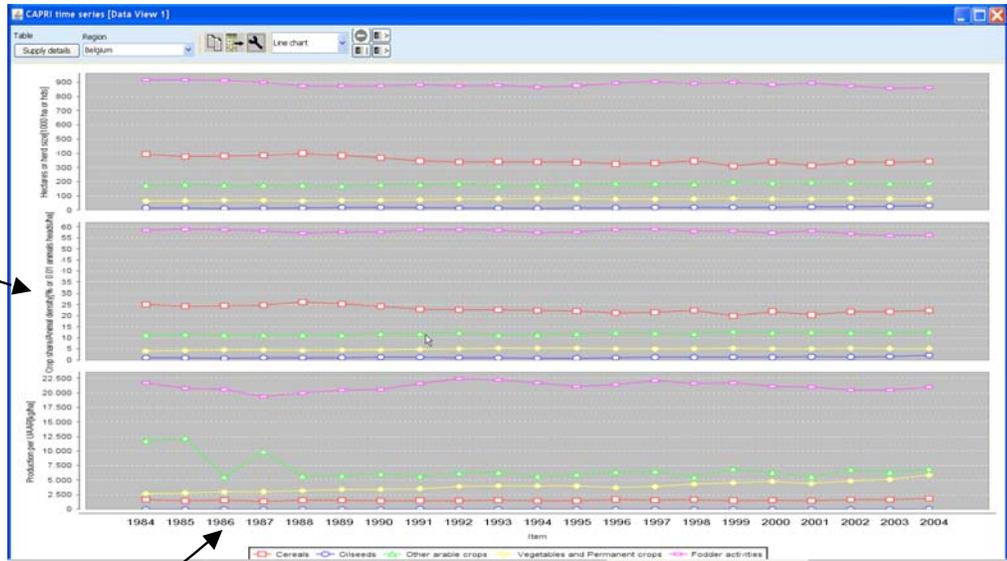
## Line and point charts

Line and point charts assume that the columns of the table present some ordered sets a e.g. years or iterations. There is currently a maximum of 25 such observations set. The different series to plot are taken from the table rows. If different column groups are present, those receive their own plot with an own value axis.

CAPRI time series [Data View 1]

Table Region  
Supply details Belgium

Year	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004				
Cereals	15.84	15.39	11.15	14.85	15.17	15.36	15.55	15.73	16.22	15.98	11.19	13.38	13.76	17.15	17.27	16.15	17.76	20.17	24.93	26.16	1.28	3.88	5.88	1.16	
Other arable crops	175.10	168.38	179.24	175.28	175.28	175.28	175.28	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13
Oilseeds	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40	64.40
Vegetables and Permanent crops	29.48	29.57	29.66	29.75	29.84	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93	29.93
Fodder activities	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
All arable crops	186.72	186.44	179.24	175.17	175.28	175.28	175.28	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13	162.13
Non arable crops	271.20	268.98	269.17	270.26	271.21	272.16	273.12	274.07	275.02	275.97	276.92	277.87	278.82	279.77	280.72	281.67	282.62	283.57	284.52	285.47	286.42	287.37	288.32	289.27	290.22



Options for line and area charts

Maximal number of plots: 5

Maximal number of series: 10

Maximal number of observations: 10

Foreground transparency in %: 0

3D effect

Plot vertical

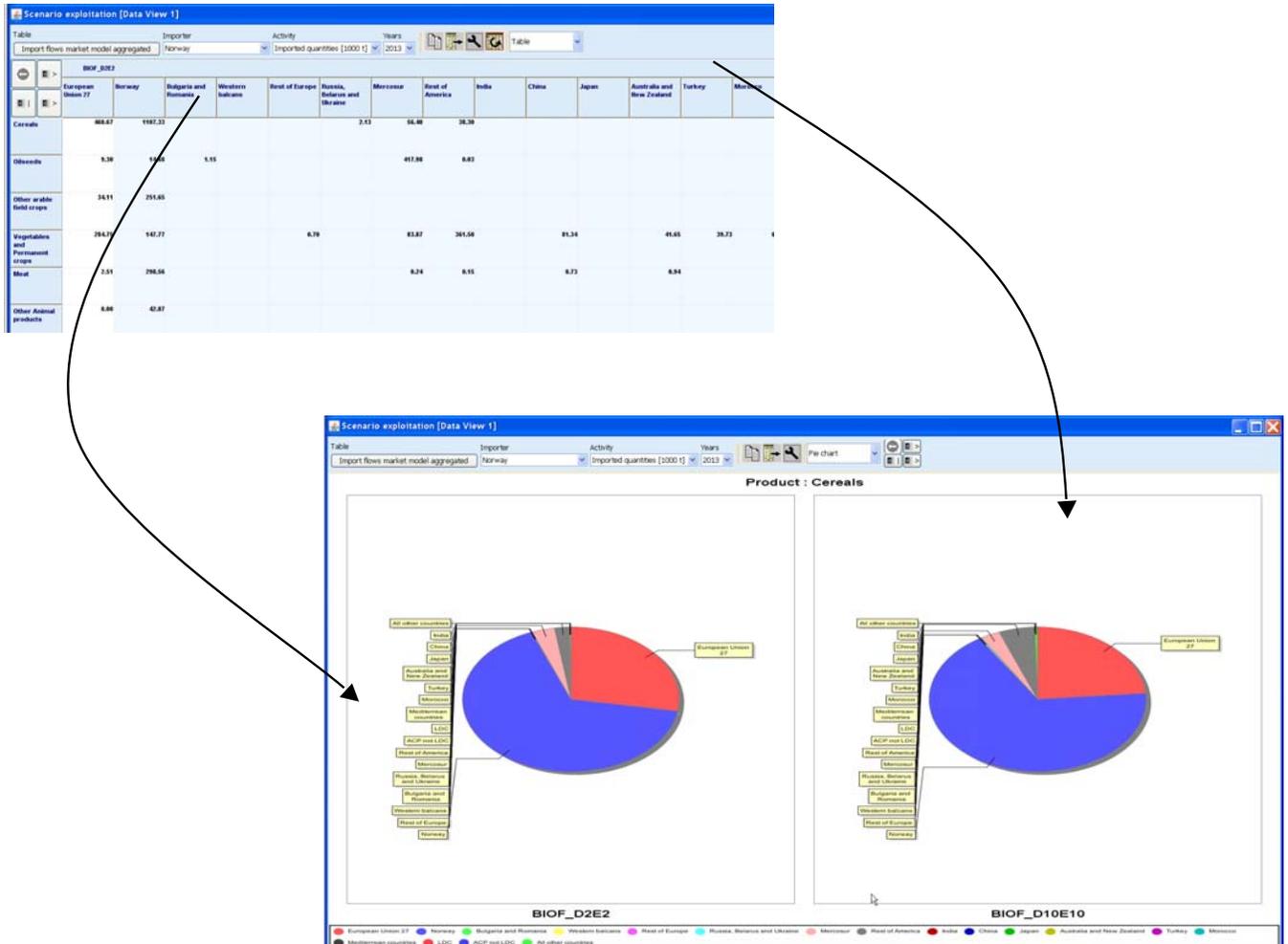
Draw lines

Draw Shapes

The options for line and area charts are similar to the ones for bar charts. The number of plots refers to the column groups, the number of series to the rows of the table. Area charts are equivalent to stacked bars, i.e. the observations are added. The number of observations is linked to the columns.

## Pie charts

Pie charts are useful to show shares on total as e.g. trade flows. The shares are calculated from the columns, whereas each column group – typically scenarios – receive its own pie. Only one row is allowed.



The user has the following options to modify the presentation of pie charts:

**Options for pie charts**

Maximal number of plots:

Maximal number of observations:

Minimum percentage to draw label:

Foreground transparency in %:

3D effect

Labels inside of pies

Circular pie

The maximum number of plots refers to the number of elements in the column group dimensions, the example above showing two plots. The number of observations defines the numbers of pies – if more

columns are available, the cake will give eventually a wrong impression is not all values are used to define the sum and the shares.

The “minimum percentage to draw label” defines a lower cut-off limit, if a cake’s size is below the threshold, no label will be drawn. As shown in the example above, setting the threshold to 100% will erase the labels (see Pie chart maps for an example). It is also possible to place the labels in the pies, and not outside of the cake as shown in the example above.

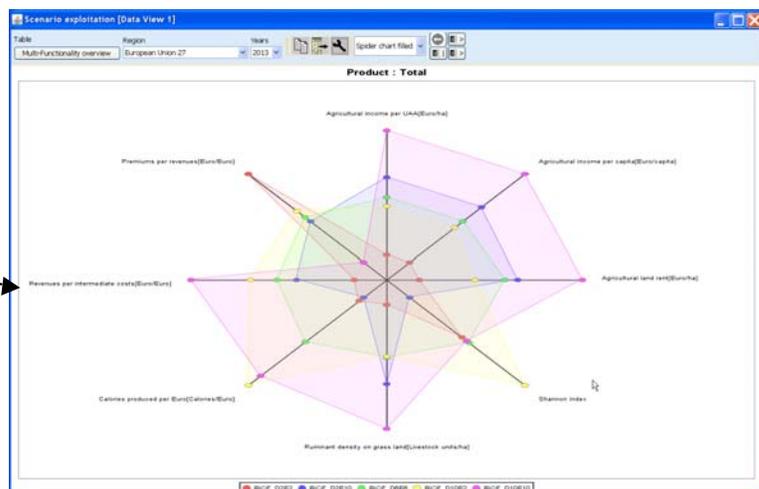
## Spiderweb plots

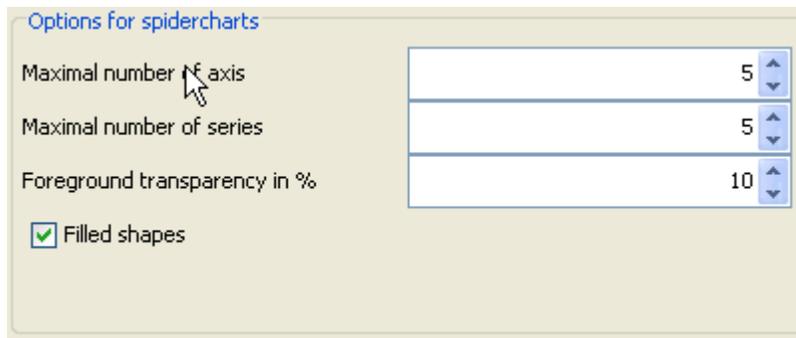
Spiderweb charts are useful to compare several dimensions simultaneously across a range of alternatives. It is assumed that the columns show the items which each receiving its own axis, whereas the column groups are the alternatives to compare. The axis are not ticked with numerical values, instead they are always scaled to cover the minimum and maximum found in any alternative.

Scenario exploitation [Data View 1]

Table: Multi-Functionality overview, Region: European Union 27, Years: 2013

	BIOF_D2E2							BIOF_D2E10					
	Agricultural income per UAA [Euro/ha]	Agricultural income per capita [Euro/capita]	Agricultural land rent [Euro/ha]	Shannon index	Ruminant density on grass land [Livestock units/ha]	Calories produced per Euro [Calories/Euro]	Revenues per intermediate costs [Euro/Euro]	Premiums per revenues [Euro/Euro]	Agricultural income per UAA [Euro/ha]	Agricultural income per capita [Euro/capita]	Agricultural land rent [Euro/ha]	Shannon index	Ruminant density on grass land [Livestock units/ha]
Total	1095.11	411.00	272.37	2.79	0.95	18.47	2.37	0.13	1138.18	427.16	292.75	2.78	0.





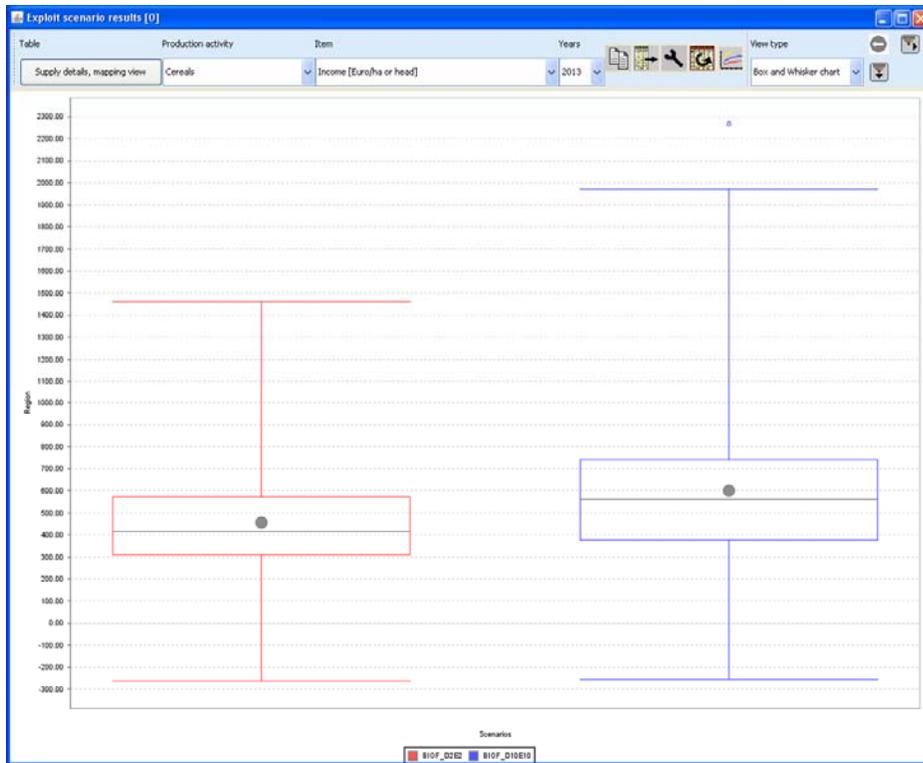
The options for spidercharts are rather limited. The user can determine, how many axis – taken from the columns – are included in the diagram, the maximum number of series, typically scenarios.

## Box and Whisker charts

In descriptive statistics, a box plot or boxplot (also known as a box-and-whisker diagram or plot) is a convenient way of graphically depicting groups of numerical data through their five-number summaries (the smallest observation (sample minimum), lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation (sample maximum)). A boxplot may also indicate which observations, if any, might be considered outliers.

Boxplots can be useful to display differences between populations without making any assumptions of the underlying statistical distribution: they are non-parametric. The spacings between the different parts of the box help indicate the degree of dispersion (spread) and skewness in the data, and identify outliers. Boxplots can be drawn either horizontally or vertically (text so far from Wikipedia).

The box and whisker charts uses the rows as the observations, and generates an own graph per column. The box show the +/-25% of the observations around the median which is shown as a grey line, whereas the arithmetic mean is shown as a grey circle. The whiskers show the median +/- three times the inner quartile range. Mild outliers are drawn as dots, and strong outliers indicated by arrows. So far there are no specific options for that type of diagram.



## Histograms

As for whisker charts and statistics shown in tables, the observations are taken from rows, and different columns are charted individually. So far there no specific options for that type of diagram. Please note that it is also possible to generate a separate Histogram window, but then, the observations refer to all columns simultaneously.



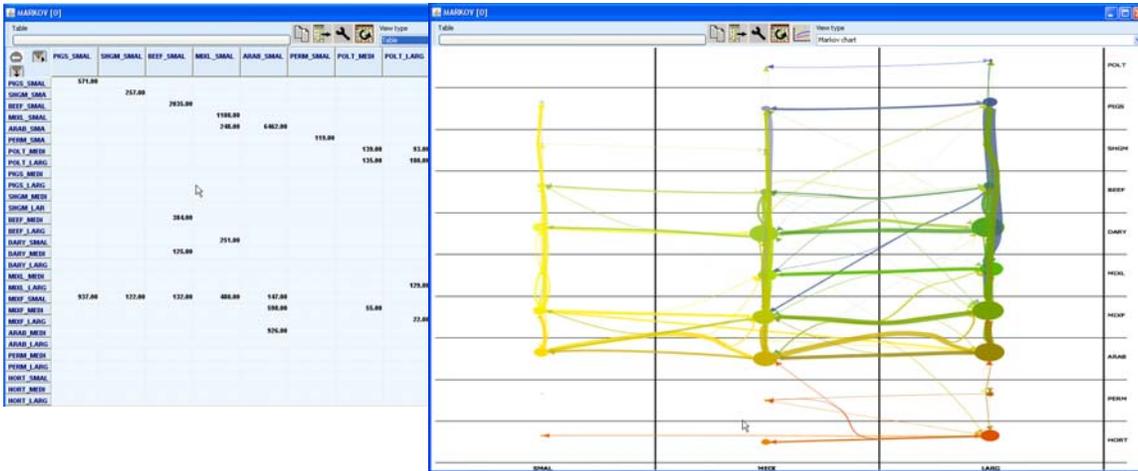
## Markov charts

A still explorative type of graphics visualizes flows between entities which are placed in a two dimensional co-ordinate system. It is currently not yet used in CAPRI itself, but applied to show flows between farm groups classified by economic size and specialization. As with the flow maps below, the major code based for the graphics is based on work of Doantam Phan<sup>1</sup>.

The positions on the x and y co-ordinate are deducted from the codes, taken from a specific section of the underlying XML-definitions which refers to a matching of sub-strings of the codes and x respectively y positions. The size of the dots is taken from the diagonal elements.

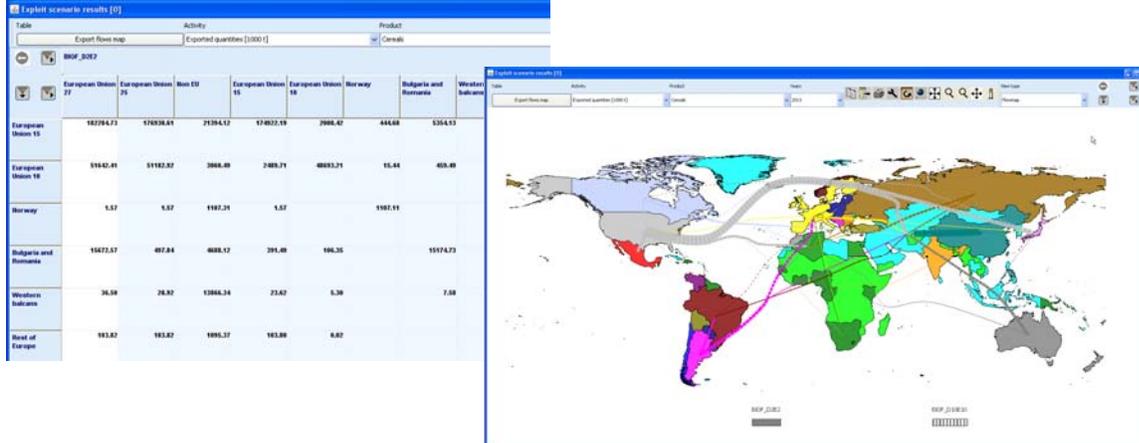
---

<sup>1</sup> Flow Map Layout, Doantam Phan, Ling Xiao<sup>1</sup>, Ron Yeh<sup>1</sup>, Pat Hanrahan, and Terry Winograd, Stanford University, see [http://graphics.stanford.edu/papers/flow\\_map\\_layout/flow\\_map\\_layout.pdf](http://graphics.stanford.edu/papers/flow_map_layout/flow_map_layout.pdf). I would like to thank Doantam Phan for letting the CAPRI team use and modify his source code.

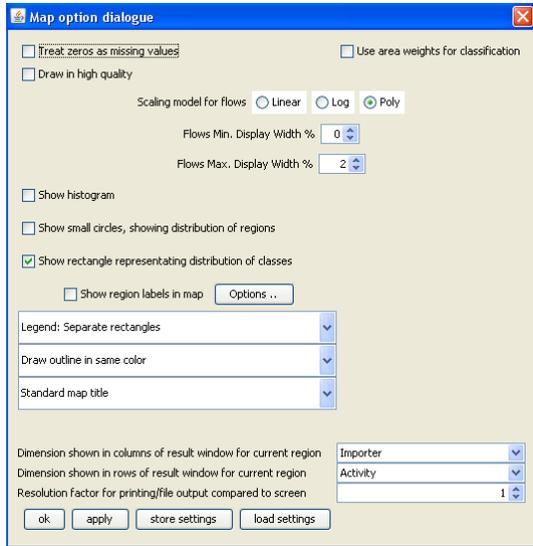


## Flow maps

Flow maps visualize flows between regions - The maps are constructed by taken the elements in the rows as the origins of the flows, and the elements in the columns as the destinations. Flows from the same origin are drawn in the same color, the width of the flows relates to their size. Counterfactuals are taken from the column groups and receive a specific “dash”. The picture below shows a screen shot of a flow map for two scenarios.



When pressing the he map option button , the following dialogue is openend:



The main options of interest for flow maps are the scaling model and the display width. The following *scaling models* are available:

- *Linear*: the width is determined by relating the flow quantity to the sum of all flows for the same scenario.
- *Log*: the width is determined by multiplying the log of the relation between the flow quantity and the minimal flow with the log of the relation of the maximal and minimal flows for the same scenario.
- *Polynomial*: the relation between the current flow and the maximal flow is raised to a power determined by taken the log of the relation between the maximal and minimal display width divided by the log of the regional between the maximal and minimal flow.

The user can prevent small flows being draws by setting a minimal width relative to the size of the window; equally, the maximal possible size of a flow relative to the size of the window can be determined.

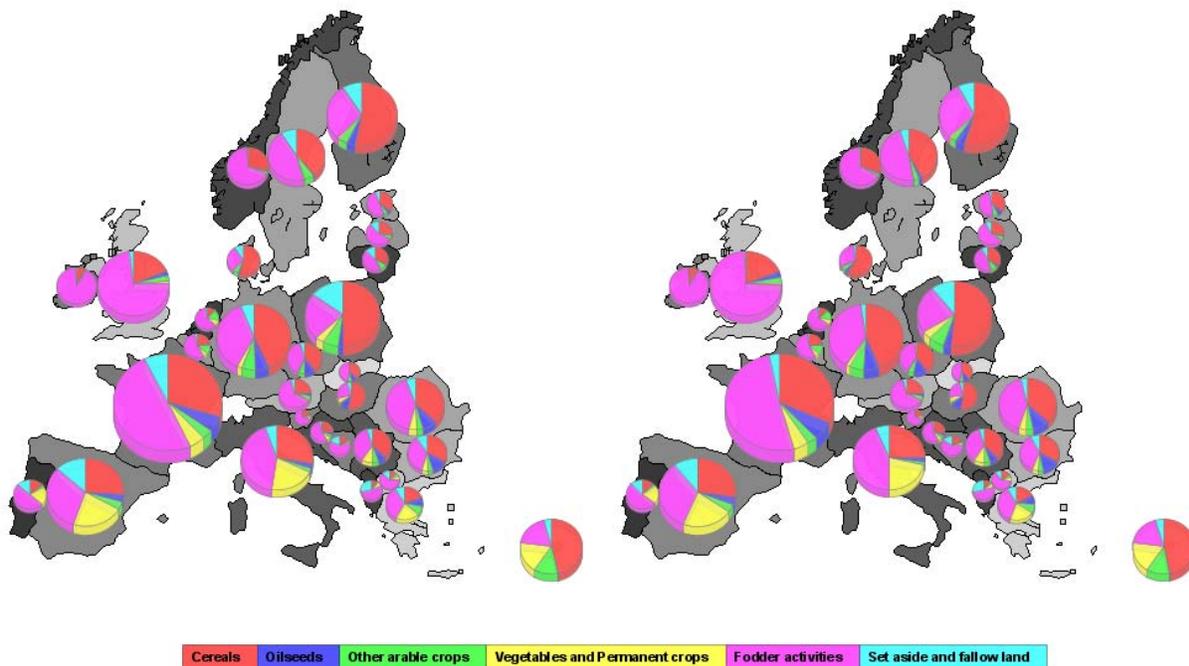


In order to show only a selection of the flows, the selection buttons can be used. The lower left one relates to the rows of the underlying tables, and thus allows excluding origins from the maps. The lower right one opens a dialogue to exclude destinations, whereas the upper right one allows exclusion of scenarios.

Most options described below for thematic maps such as zooming and dragging are also available for flow maps, however, clearly, classification and colors models are not supported.

## ***Pie chart maps***

Another less standard application of maps is the possibility to place pie charts above the geometry. The regions must – as always with maps – be placed in the rows of the underlying tables, and the cakes are calculated from the data in the columns. It is possible to produce maps for different scenarios when those are placed in the column groups as shown below. The size of the charts depends mainly on the bounds of the underlying polygon, so that smaller countries have smaller pies. The settings for pie chart diagrams (see Pie charts) can be applied to that view as well by opening the



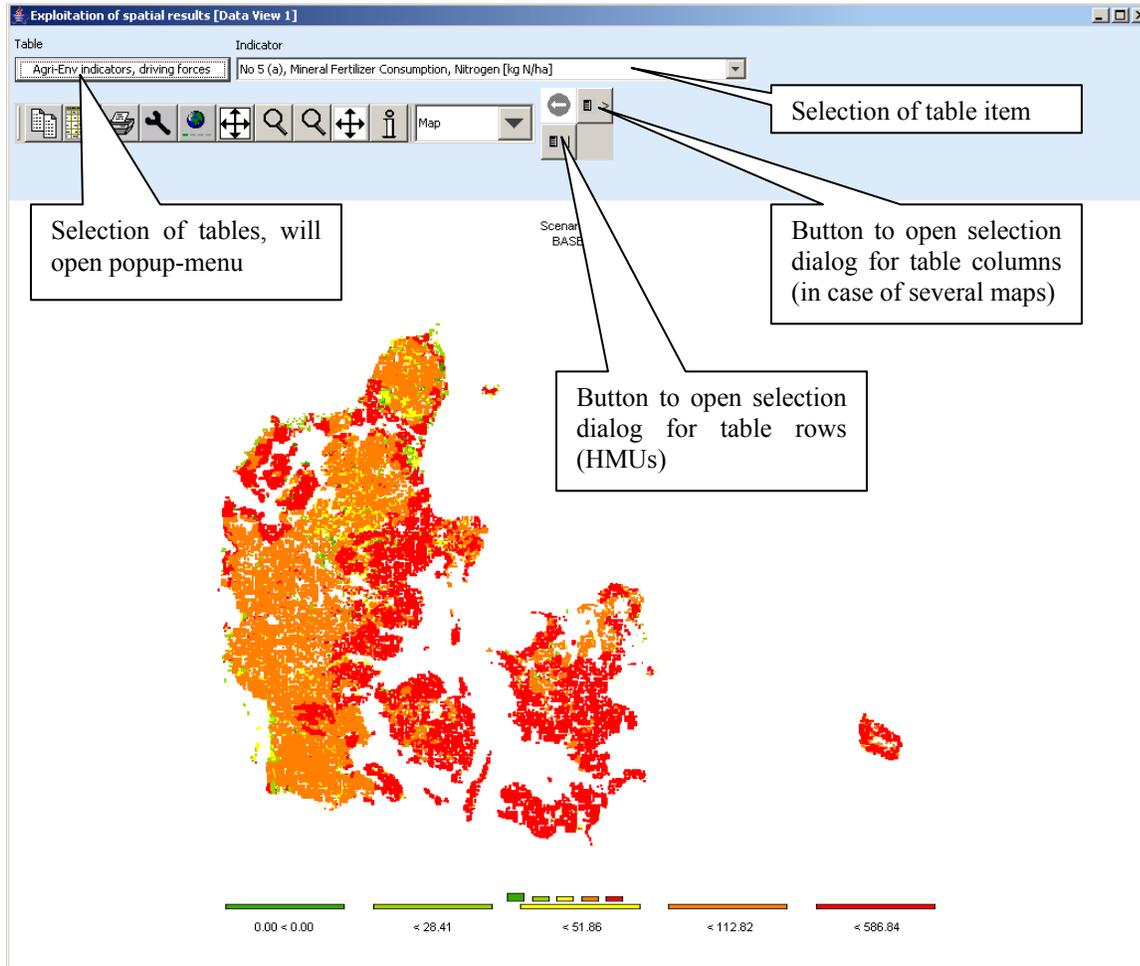
## ***Colored thematic maps***

The GUI currently provides geometries for NUTS 2 regions, Member States, the regions with behavioral functions in the market model, trade blocks in the market model and finally, the Homogenous Soil Mapping Units (1x1 km resolution) underlying the spatial down-scaling component. The geometries are always linked to the rows of the underlying table.

Both for NUTS2 results, but more so for the results at the HSMU level, the most obvious way to visualize results is the use of thematic maps. When starting the GUI, the mapping view uses some pre-sets which can be interactively changed as described below. The following screen-shot shows the result of loading

the base year results from the spatial dis-aggregation for Denmark and then switch from the tabular to the mapping view.

As with other views, the content of the map can be changed by working with the drop-down boxes, or by (de)selecting columns and rows. The map specific possibility to change class limits and colors and further features are discussed in the following.



## Changing the classification and the legend

In order to change the layout of the map, click with the mouse in the area of the legend or press the map

option button . The following dialogue will open.

The screenshot shows the 'Map option dialogue' window with several callouts:

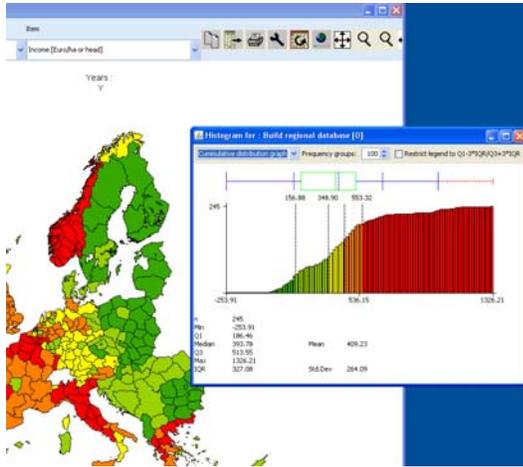
- User options to change colors and classification:** Points to the 'Classification method' dropdown (set to 'Quantile') and the 'Number of classes' spinner (set to '5').
- Current class definitions (labels, limits, colors):** Points to the table below the 'Color table' section.
- Click to show histogram window, showing current class limits and colors:** Points to the 'Show histogram' checkbox.
- Options for output in map window:** Points to the 'Show small circles, showing distribution of regions' and 'Show rectangle representing distribution of classes' checkboxes.
- Options for info window:** Points to the 'Dimension shown in columns of result window for current region' dropdown (set to 'Years').

#	label	class limit	% of obs	color
1	-253.91 < 156.88	156,884	20	Green
2	< 348.90	348,901	20	Yellow
3	< 446.65	446,653	20	Red
4	< 553.32	553,321	20	Orange
5	< 1326.21	1,326,205	20	Red

It offers different options to change the way the map is drawn on screen and information supporting the classification.

## Adding a histogram window to a map

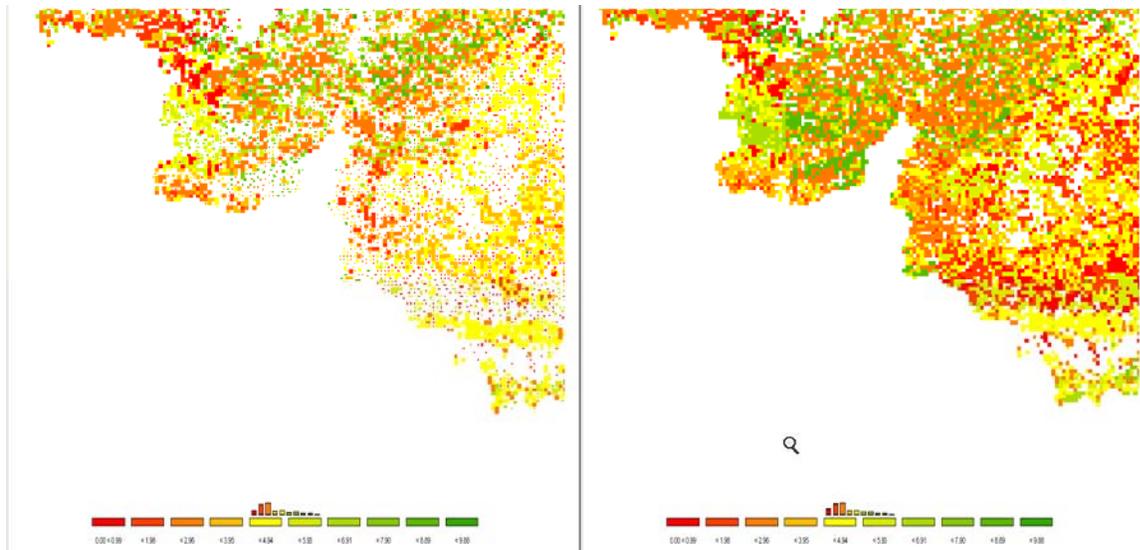
In the map option dialogue, tick “Show histogram” and a separate window with a Histogram will be shown. It will use the current classification and color model to visualize the distribution of the values, reports some basic statistics and shows a box and whisker diagram.



## Shrinking polygons according to UAA share

The optical impression received from a map where colors are used to differentiate between values is to large extent depending on the area covered by certain color. If the majority of the pixels is drawn in red, that will send a warning message to the user. In the case of the HSMUs and information relating to agriculture that message can be strongly biased as almost all HSMU comprise some other land cover than agriculture, and some of the HSMU comprise only very little agriculture, but e.g. forest, shrub lands, water bodies or artificial land cover. The HSMU geometry therefore comprises the information about the share of UAA assigned in the base year to each HSMU, and that information can be used to shrink the area of the polygons when drawn on screen accordingly. That is done by drawing all points of the polygons towards to the centroid of the polygon, by multiplying the distance between the point and the centroid with the square root of the share of the UAA. In the original HSMU geometry, such polygons had been broken down to simpler ones where the connection between a point and a centroid would cut through a segment of the polygon a in such case, shrinking could let the new polygon to hide other ones.

The graphs below show the very same map (same input data, classification and coloring) for the High Nature Value indicator for a part of Belgium. The right hand side map draws the HSMUs in their full size, the left hand side one using shrinking. The message perceived is probably mighty different. In the unshrunk right map, one may conclude that there is a lot of highly intensive agriculture (low HNV indicator drawn in red) in the lower diagonal triangle, and some important areas of high nature farmland in the protruding area. That optical impression is different with the polygons are drawn corrected for the share of agriculture. It turns out that in lower diagonal triangle, the density of agriculture is often low, and especially low in the intensively managed HSMUs. Equally, it turns out, that the area covered by High Natural Farmland in the protruding part is relatively small.



## Area weighted classification

The classification can be generally applied treating each “region” (a NUTS II or a HSMU) as an observation with equal weight or using the areas of the underlying polygons as weights. Those weights are multiplied with the share of UAA if shrinking is used as explained above.

## Excluding zeros from classification and removing small and large values

In GAMS, zeros and missing values cannot be distinguished. For certain results, zero results are therefore coded as very small numbers to allow for that distinction. Zero observation can be excluded from classification and the polygons with zero observations will not be filled. Equally, a number of regions with small and large values can be excluded from classification.

## Classification method

A first important feature is called “classification method” and defines how internally the class limits are set. For all types of automatic classification methods a clean up procedure is used which removes classes with identical limits. It is generally recommended to use a number of classes which can be easily identified by the user, and to consult the frequency or cumulative distribution graphs present in the map option dialogue to check to what extent the class limits chosen represent well the data.

The following classification methods are currently supported:

## Natural breaks

Natural breaks classification is a method to cluster the data into classes so that differences between the means of the classes become high while the standard deviation inside the classes becomes low (FISHER, W. D. (1958). "ON GROUPING FOR MAXIMAL HOMOGENEITY," JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION 53, 789-798. Code based on: HARTIGAN, J. A. (1975). CLUSTERING ALGORITHMS, JOHN WILEY & SONS, INC., NEW YORK. PAGES 130-142.). The algorithm does only find the approximate best solution, but gives often a rather appealing class limit definitions.

It works rather well if no extreme outliers are present in the distribution. In the latter case, classes solely comprising the outliers will be generated, and the vast majority of the values will be put in one or two classes.

The clustering algorithm is rather expensive to calculate, so that in cases of population exceeding 500 observations a somewhat simplified version is implemented in the CAPRI GUI. From the original observations, a “condensed” population is generated whose members represented means of consecutive observations of the original one. The members are set so that the number of observations from which the mean is calculated is not bigger than 1/500 of the original population size and that the spread of those observations is smaller than the minimum of 1/500 of the spread of the total population and 10% of the standard deviation. The actual calculations are then done taking the size of the resulting classes into account.

## Quantile

The observations of the regions are split such that in each class fall approximately the same number of observations. Quantiles are cheap to calculate and are therefore the default setting, and often appealing as colors occupy similar areas in the maps as long as the polygons have approximately the same size.

If unique values are found at the end of a quantile, the algorithm will either exclude all observation with that unique value from the class or include all of them. The decision will be based on the fact if with or without inclusion the size of the class comes closer to the desired size. If the user has e.g. chosen five classes, the desired class size should cover 20% of the observations or area weights.

## Equal interval

The differences between the current minimum and maximum value is divided into classes of equal spread. May lead to rather curious class limits when outliers are present. In those cases, it may be appropriate to

exclude some regions from the classification. See below for details how to exclude regions from the classification.

## Mean standard dev

The class limits are defined according to the mean and portions of the standard deviation of the data. Works best with normally distributed data, but may result in very small classes if the distribution is skewed, e.g. long tailed. The algorithm will always introduce at least four classes, then six, eight, ten and twelve. More than twelve classes are neglected.

The algorithm takes into account the spread of the data, and sets the class limits accordingly. If all observations fall into +/-25% of a standard deviation, class limits are introduced at 25% and 10% for four classes. If the number of classes is higher, new limits are introduced at 5%, 2.5%, 1% and 0.5%. In case of +/-50%, the smallest class is dropped and +/-50% added, and so forth up to +/- 3 standard deviations.

## Nested mean

The nested mean classification will only work with 2, 4 or 8 classes. The classes will be defined such that one break is found at the mean of sample. The resulting two halves of population are then again divided by their mean to get four classes, and the resulting quarters divided by their means to define eight classes. Works well with rather skewed distributions.

## Manual classification

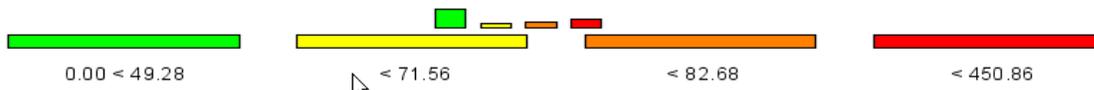
Finally, the user may set the class limits by hand. In order to do so, double click with the mouse on the appropriate row in the table with the classification results in the column "class limit". The value can now be changed with the keyboard. When done, click into another cell. The labels will be adjusted accordingly. Afterwards, when all the class limits are defined, the user may also overwrite the label (e.g. using words as "low" or "high").

Please keep in mind that currently the values will be lost if you load other data or change the classification, number of classes etc..

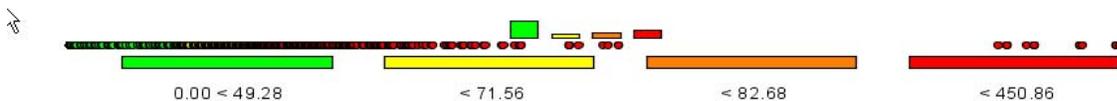
#	label	class limit	% of obs	color
1	0.00 < 0.00	0	30,488	
2	< 28.41	28.407	17,398	
3	< 51.86	51.86	17,398	
4	< 112.82	112.821	17,398	
5	< 586.84	586.835	17,317	

## Integration distribution information in the map window

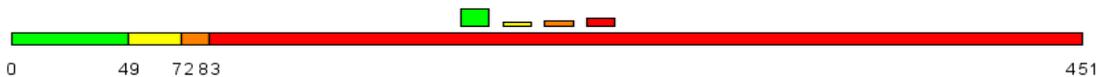
The GUI allows the user to enter in different ways distribution information in the map. The first possibility is to print a simple frequency diagram above the legend. That gives a rather intuitive feel on how well the class limits represent the data distribution. In our example below, it is obvious, that the majority of the values in the first class.



Less suitable for final out, but useful while playing around with classification methods and class definition are the distribution dots which can be added. They carry additionally information about the where the values of in different classes can be found.



Finally, switching to linear or logarithmic may be a way to help reading the map.



## Color table

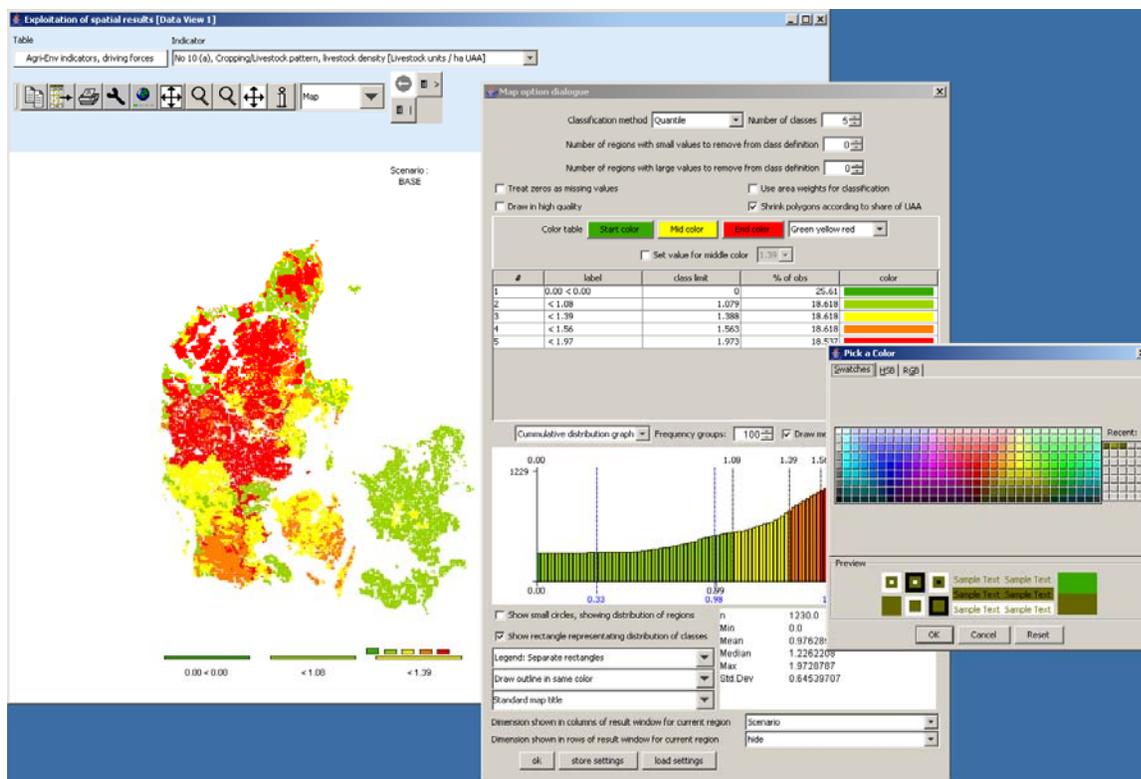
The color table defines the colors used for the classes. When choosing the color model, keep in mind that colors carry a meaning; red e.g. is generally interpreted as dangerous. Equally, it is important to think about the final medium with which the map will be published. Exporting colored maps to a black-white device will render it almost impossible to read the map. Is it best to try different color tables and different classification methods on your data. The following color models are currently available, named according to the data order from minimal to maximal value:

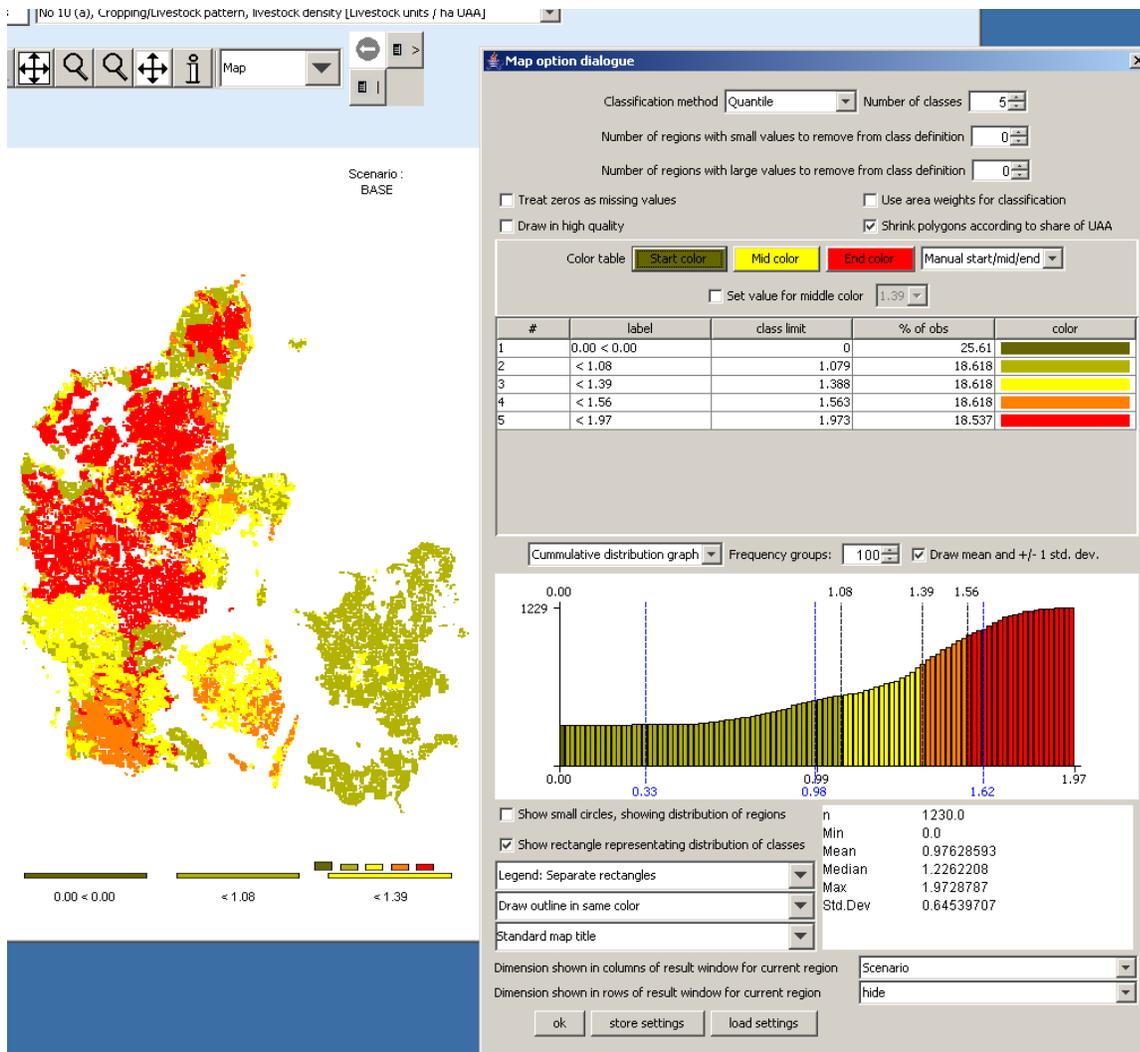
- *Green – Yellow – Red (standard)*. Normally, the middle class is drawn in yellow, smaller values in shades between yellow and green, and larger ones from green to red. Should be applied e.g. to environmental indicators where the damage increase with the value of the indicator.
- *Red – Yellow – Green*, as above, only that high values are shown in green. Should be used e.g. for income indicators or environmental benefits.

- *Red – Gray - Green / Green – Gray – Red*, more available for historic reasons as they mimic the color tables of the original JAVA applet.
- *Blue – Gray - Green / Green – Gray - Blue*; introduced on demand of DG-AGRI. A good choice if the “good”/”bad” interpretation of the distribution is to be avoided.
- *Shades of grey*; sometimes needed for publications when color printing is available in the final hardcopy. Beware to use a limited number of classes.
- *Shades of blue*; useful where the notion or “bad” or “good” inheritably comprised in greenish and reddish colors is to be avoided.

## Defining an own color model

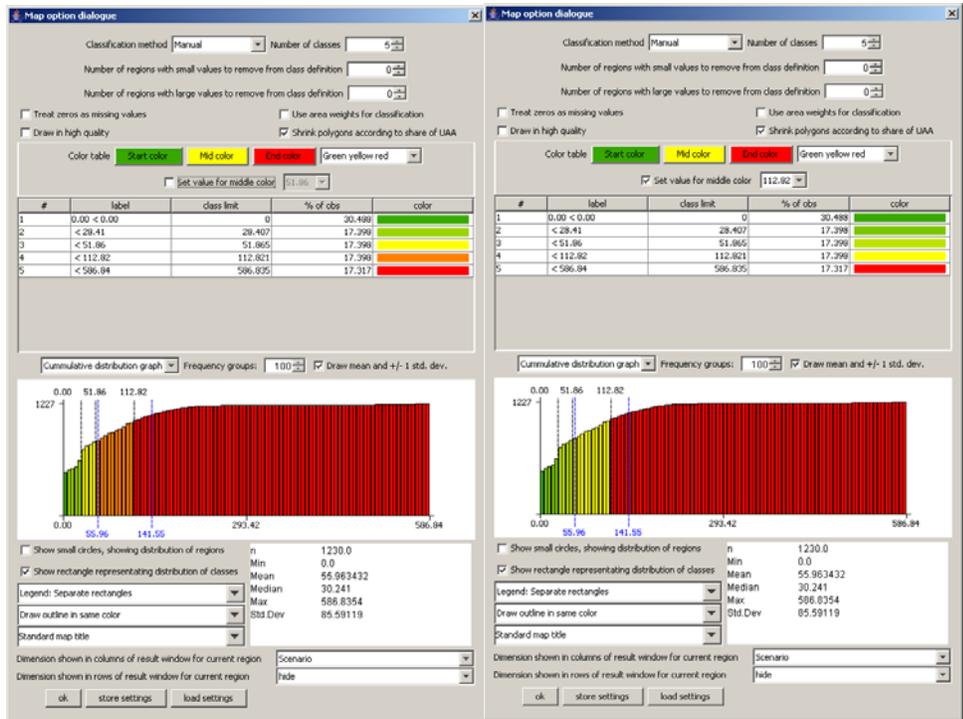
Once a color model is chosen, the user can re-define the start, middle and end color using the three buttons below the color table selection row, as shown below, given a lot of freedom to generate color ramps.





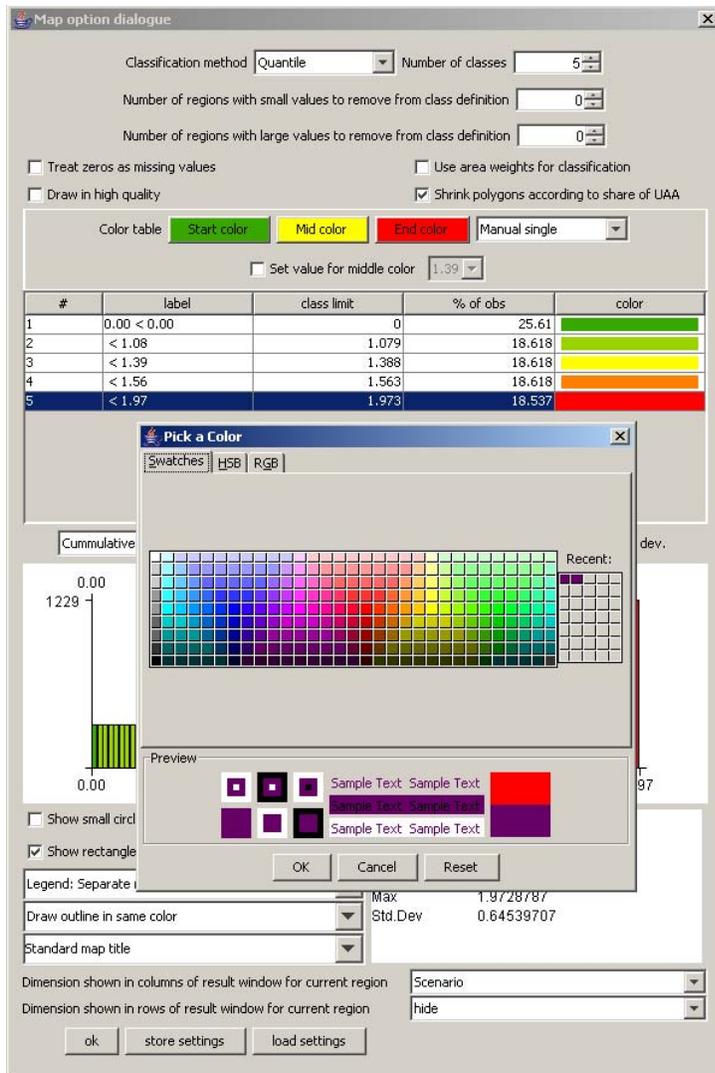
## Changing the value for the medium color

Normally, the medium color (yellow or gray) is assigned to the middle class. Sometimes, the user may wish to change the class where the color switches. First, the “Set value for color change ..” must be ticked on. Next, in the now enabled drop-down box, choose the class limit for which the middle color should be used. The effect is shown below. Whereas before, values in the class below “392.70” – the middle class – was drawn in yellow. When the user now selects another class limit, the colors assigned to the classes change, here one of the shades of green is dropped and shade of red is added.



## Manual set colors

Finally, the user can choose its own colors by double clicking in a color field in the legend table. That should only be done after the final definition of the class limits is set as otherwise, the manually set color will be lost.



## Changing the way the legend is drawn

The map viewer puts the legend always below the map, and currently offers three options how legends are drawn:

1. Separate, equally sized rectangles which show the upper class limit with the exemption of the lowest class, which shows the lower limit.



2. A continuous linear scaling bar. That gives an optical idea about the how distribution of the class limits. Overlapping of the number of avoided.



### 3. A continuous logarithmic scaling bar .



In all the cases, the tool dialogue can be used to set number of digits shown, e.g. reducing the number of digits to zero leads to a linear bar as shown below:



The reader is reminded that the label can be changed manually as shown below.

**Map option dialogue**

Color table: Green yellow red  
 Set value for color change for Green yellow red: 345.89148  
 Classification method: Quantile  
 Number of classes: 2  
 Number of regions with small values to remove from class definition: 0  
 Number of regions with large values to remove from class definition: 0  
 Legend: Separate rectangles

#	label	class limit	color
1	low	345.891	yellow
2	high	1,392.846	red

Cumulative distribution graph: Shows a distribution curve with a vertical line at 345.89. X-axis values: -418.65, 487.10, 1392.85. Y-axis value: 250.

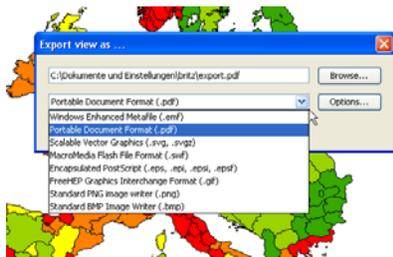
Statistics:  
 n: 249.0  
 Min: -418.65223  
 Mean: 358.84164  
 Median: 345.89148  
 Max: 1392.846  
 Std.Dev: 4468.262

Dimension shown in columns of result window for current region: Years  
 Dimension shown in rows of result window for current region: Activity

Map view: Shows a map of Europe with a color scale from yellow (low) to red (high). Legend below the map: low (yellow), high (red).

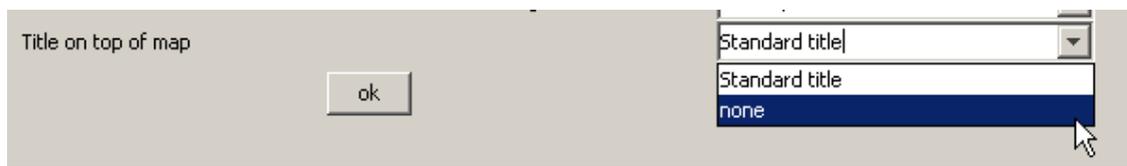
## Copying the map to the clipboard or saving to disk

In order to export the map to other applications, the easiest way is to use the clipboard, in order to do so, press the “copy to clipboard”  button. Afterwards, the current map can be imported into other applications as e.g. MS Word. Another possibility is to store the current map in jpeg format on disk, to do so, use the “export”  button which will open a file dialog to choose the name of the file and selected between different graphic formats. For MS Office users, the “Windows Enhanced Metafile (.emf)” format is especially interesting as it allows the change the graphic afterwards, e.g. by moving the legend or changing text.



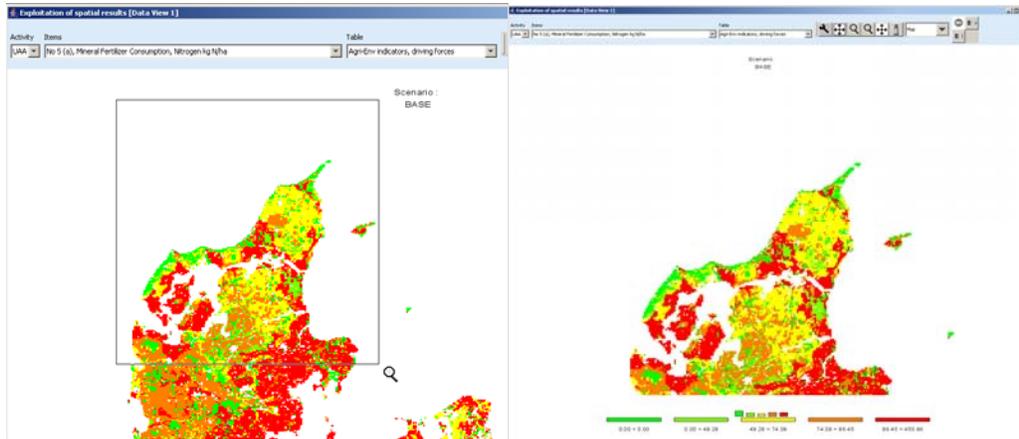
## Changing the title of the map

When using output to clipboard or disk, the user may often prefer an own title or no title at all on top of the map in order to produce a caption for the map in another application. In order to refrain from drawing a title on top of the map, click into the legend part of the map, and in the dialog at the bottom, choose “none” in the row labeled “Title on top of map”. Alternatively, the user can simply write something in the box.



## Zooming in and out and navigating in the map

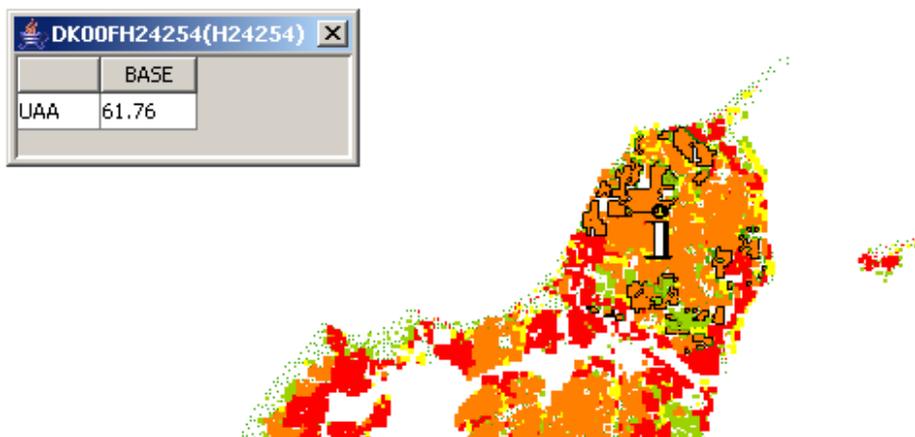
In order to zoom in part of the map, press the  button. The mouse pointer will change to a magnifying class with a cross in it. You can then mark an area on the map by pressing the mouse button, dragging and releasing the mouse. After the mouse is released, solely the selected zone of the map will be drawn, without changing the class limits or any other setting. Clicking with the mouse while in zoom in mode will increase the map resolution step-wise by 25% and center the map at the current mouse position.



By clicking with the zoom out pointer  on a point of the map, the point becomes the new center point of the map and the map resolution is reduced stepwise by 25%. Equally, you may drag the map while keeping the current resolution by choosing the drag pointer . And finally, in order to return the original full sized map, use the “full extent” button .

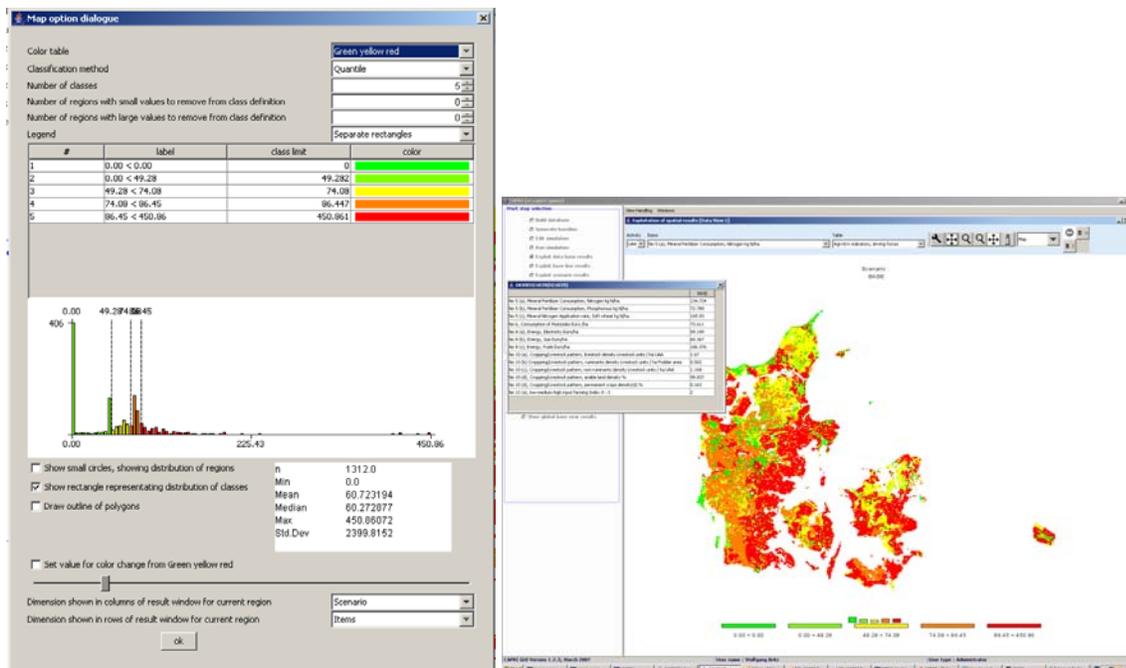
## Getting data for specific polygons

The info pointer  will open an additional window as shown below which displays information on the current polygon – the circle above the “i” being the focus point. The new window shows in his title bar the code and, if available, the long text of the polygon currently pointed to with the info pointer. The content of the info window is continuously updated when the mouse is moved over the map, and all polygons belonging to the same region as the one where the mouse is over are highlighted.



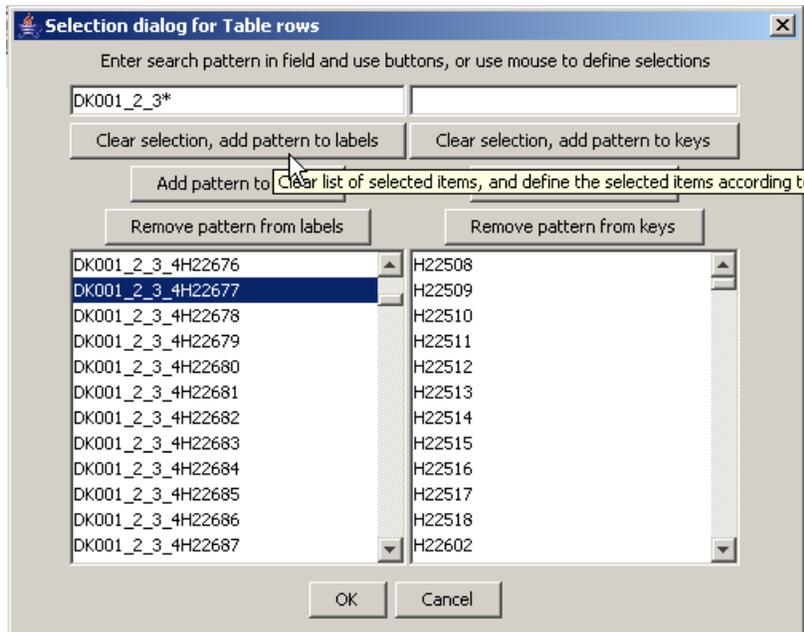
If the user opts to use one of the comparison options to be shown (percentages, differences, normalization) by clicking on “customize” button, an additional column is automatically added to the info window showing the comparison value used. That is especially helpful when the map shows only differences.

The content shown in the info window is not fixed, rather, the user can decide which data dimensions to use for the columns and rows by using the “map option dialogue” by clicking on the legend of the map. If the user e.g. switches to “items” instead of “activity”, the “info” window will look like shown below. An alternative is to use a second tabular view in addition to the map.

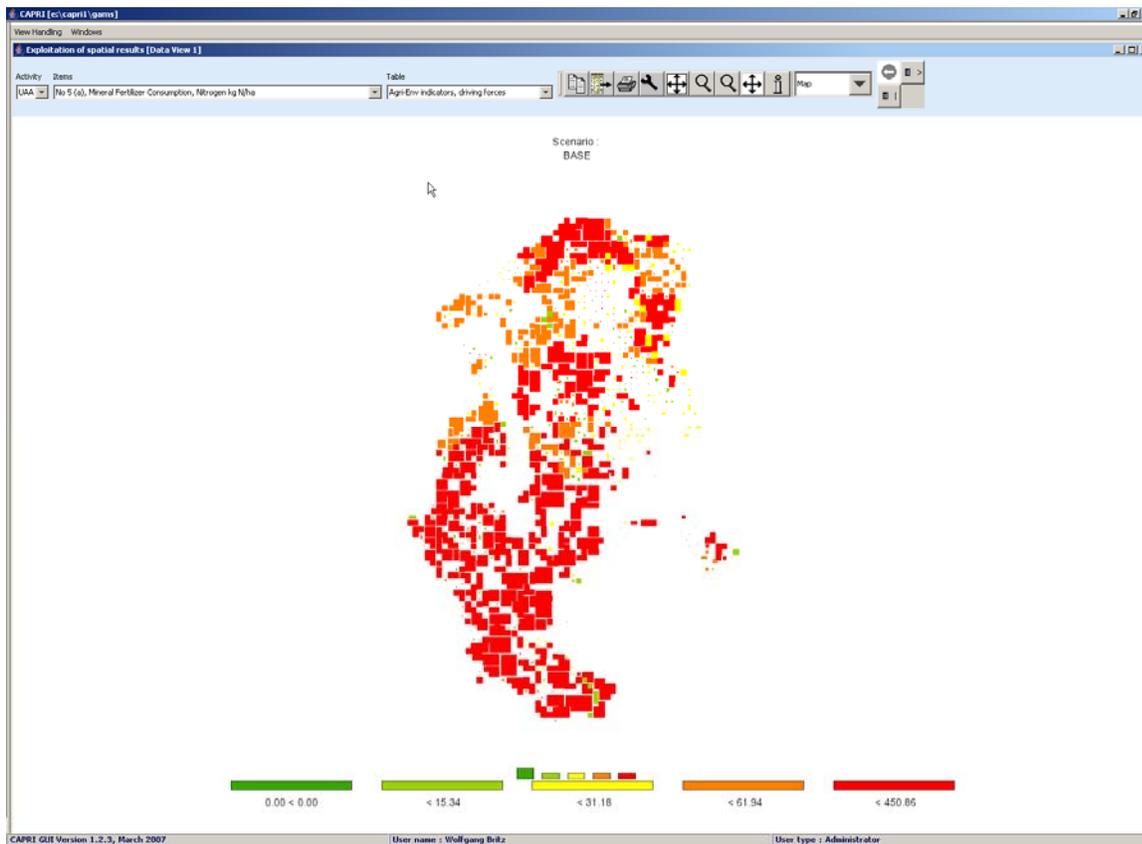


## Highlighting specific regions in the map

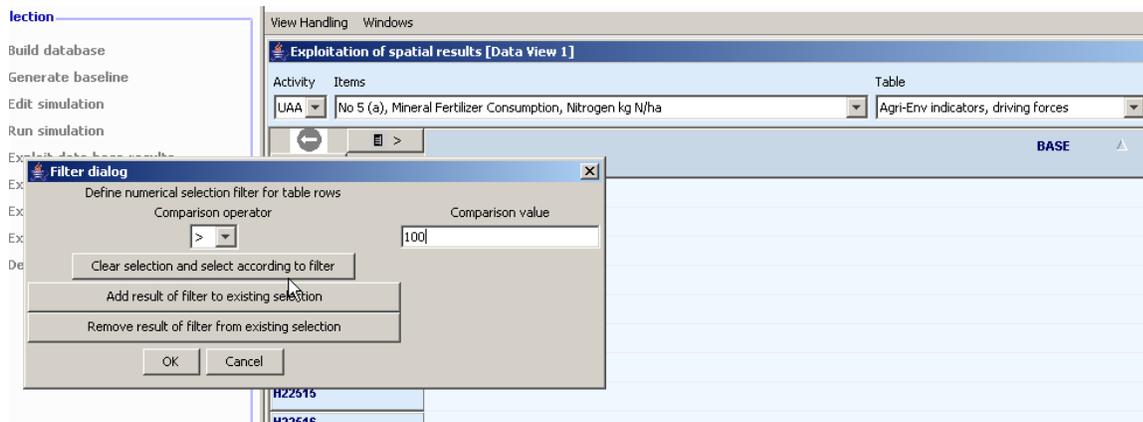
Sometimes it may be interesting to see the spatial distribution of specific data or data constellations. All views open the possibility to (de)select columns and rows, allowing e.g. to use the NUTS code in front of the numerical HSMU code to select only the HSMU belonging to specific administrative regions. That possibility is explained in short. First, press the row selection button  (“Open selection dialog for table rows”) which will open the following dialogue.



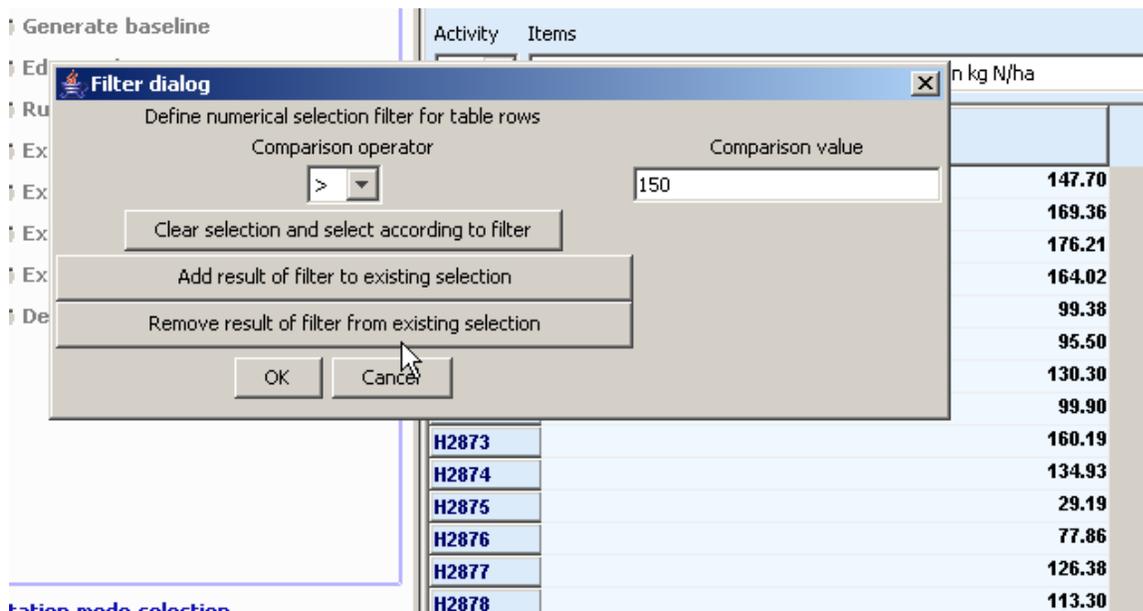
Now, we may e.g. selected only the HSMU belonging to the FSS region DK000\_1\_2\_3 by typing “DK001\_2\_3” in the left input box, and then choosing “Clear selection, add pattern to labels”. Afterwards, the map will look as shown below.



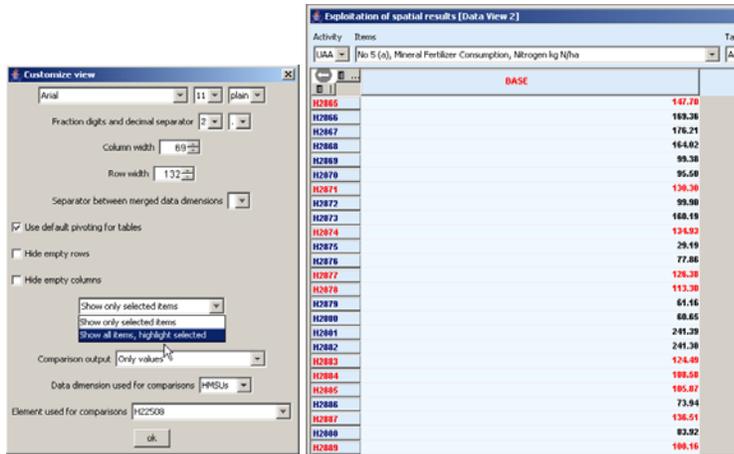
The tabular view opens up the possibility of using numeric filters, an option discussed in the following. Take as an example the task to select all regions where the Nitrogen Fertilizer Consumption is between 100 and 150 kg/ha. First, switch from map to tabular view. In the table click with the right mouse button in the column header of that column holding the values to which the filter should be applied, as shown below. We will need to apply the filter step-wise, first e.g. selecting all values greater than 100 and then removing those which are above 150.



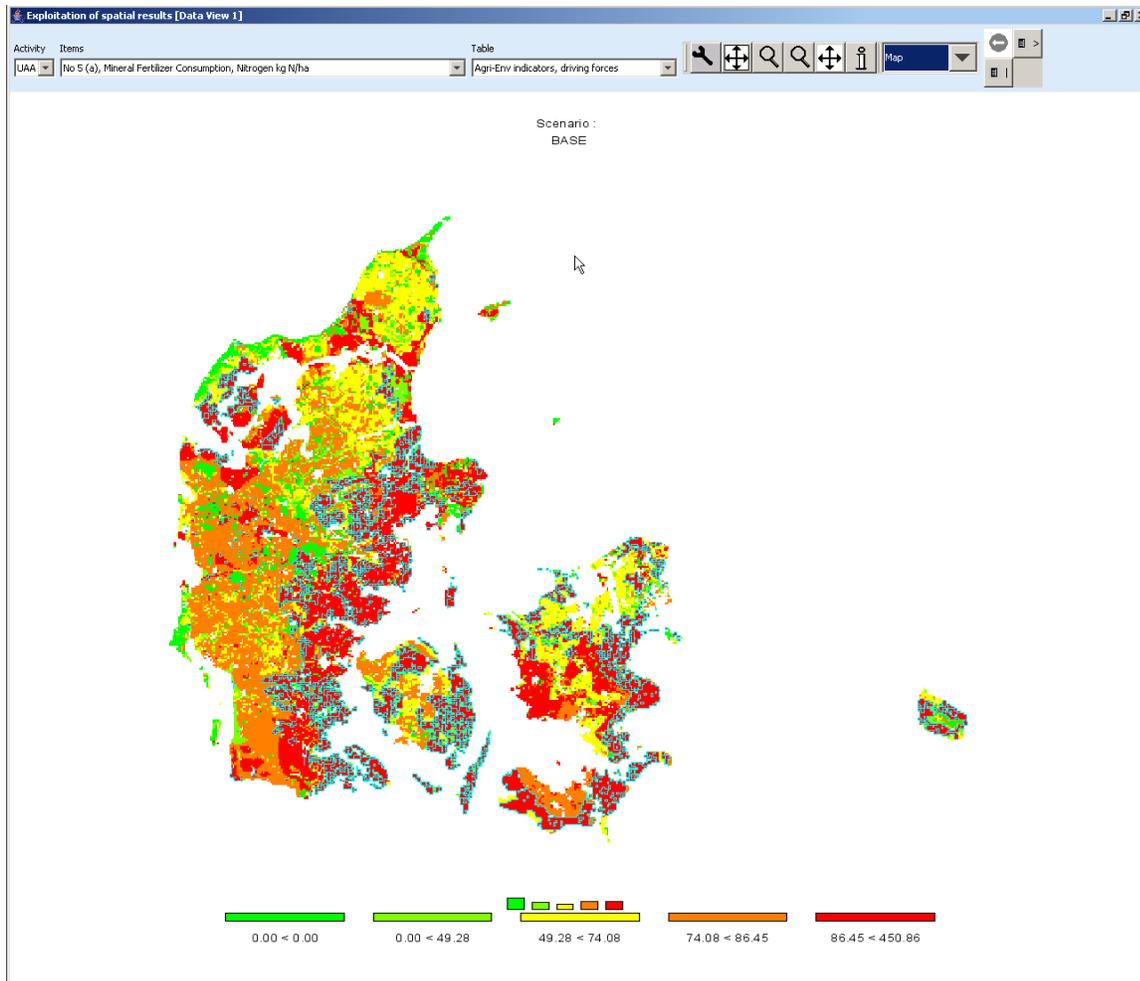
After pressing on “clear selection and select according to filter”, and then on “ok”, the table will only show such regions where the value in the column “BASE” is above 100, as shown below. Next, we must exclude the regions above 150 kg/ha. To do so, set the filter to “>” “150” and then press “remove result of filter from existing selection”.



Now, drawing a map with just those regions is not so interesting. However, with the tool dialogue, we can highlight the selected value instead of hiding all others. The selected rows are now shown in red in the tabular view.



When we now draw the outlines of the selected polygons only (see map option dialogue), the map will draw the outline of the selected regions in cyan and thus highlight them. The row selection will be maintained when the pivot or the table is changed as long as one of the selected items can be found in the rows of the new table. The example map shown below is certainly not so interested, as changed class limits could have done basically the same job. However, we could switch e.g. to grass land shares to see if that fertilizer input is more often found on arable or grass land.

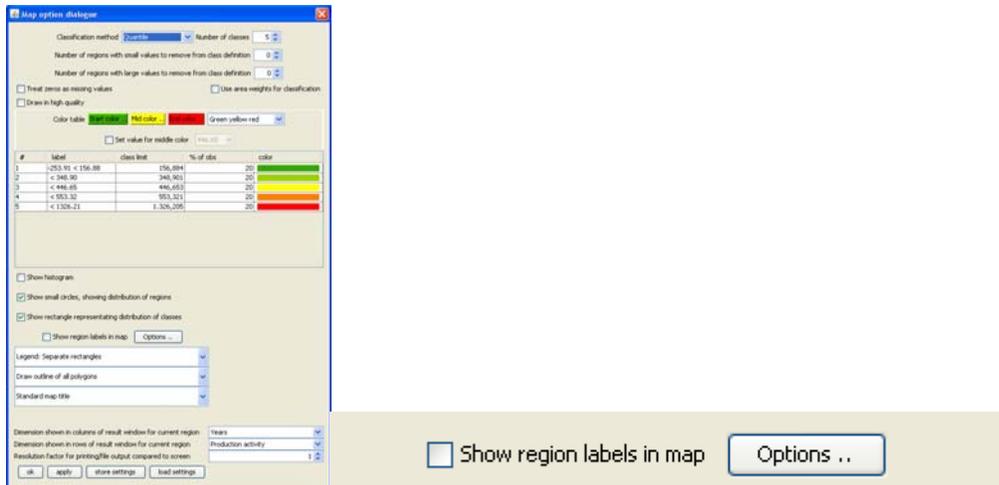


## Updating the map

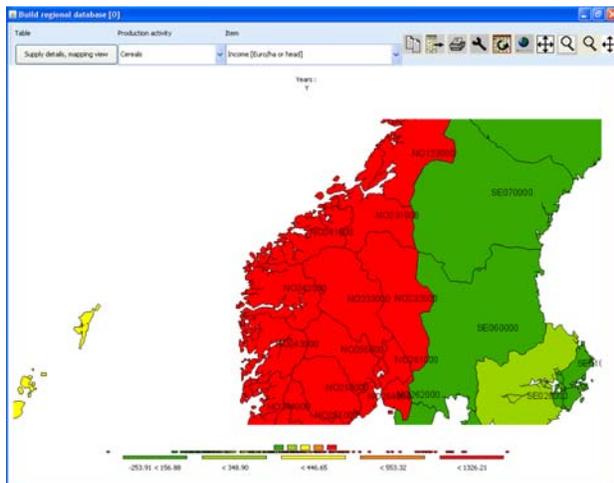
Generally, the map is updated automatically when the user changes an option impacting on its layout as long as long as the number of visible polygons is below 20.000. If that amount is exceeded, the classification dialogue is updated immediately, but not the underlying map. In order to apply the changes, the “apply” button must be pressed. The user is informed that the “ok” button will also update the map, so that an “apply” immediately before an “ok” is not necessary.

## Adding region label to the map

In the map option dialogue,



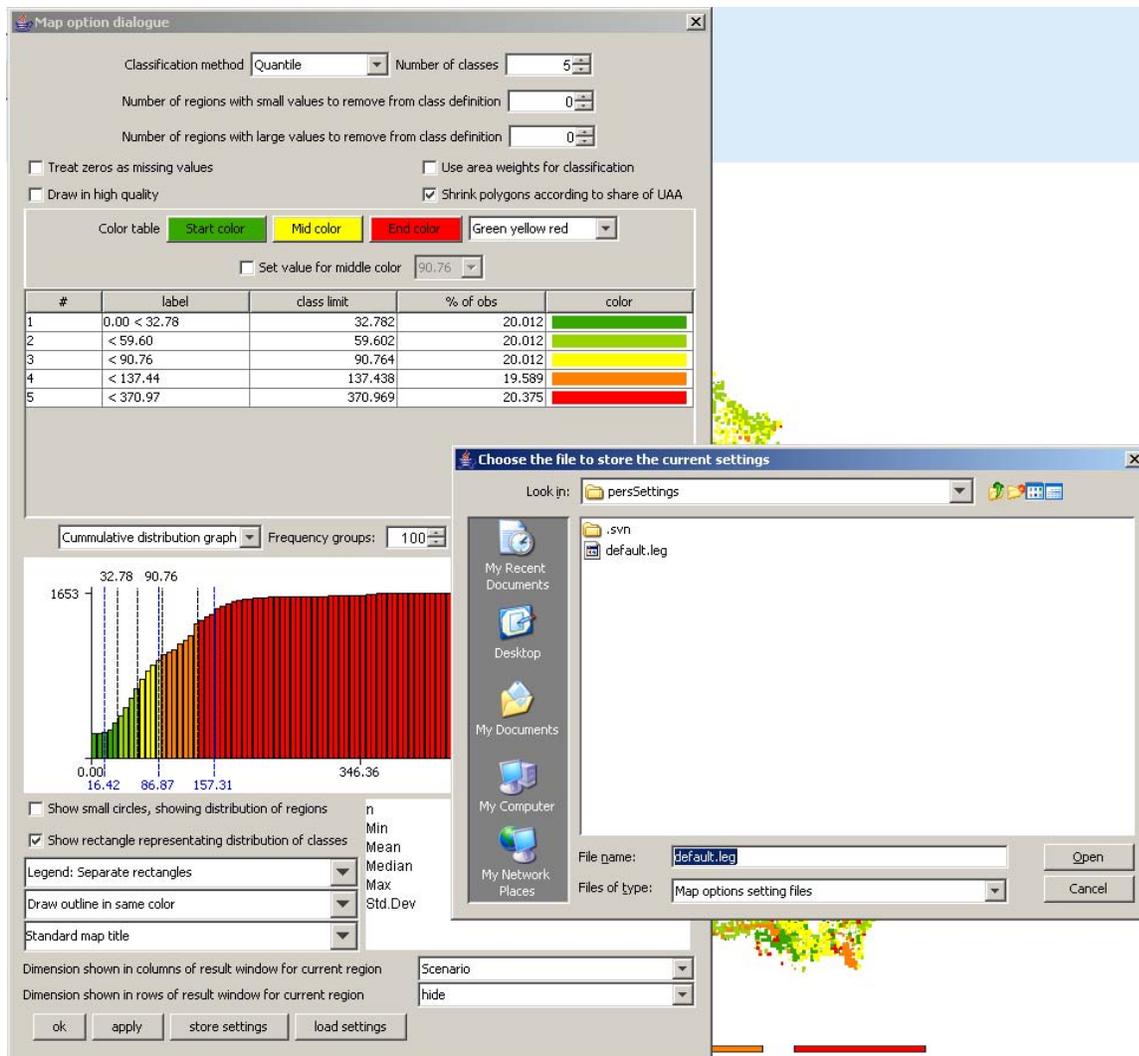
Tick the above box “Show regions labels in map” to add labels to the largest polygon for each regions as shown below.



## Storing and re-loading your settings

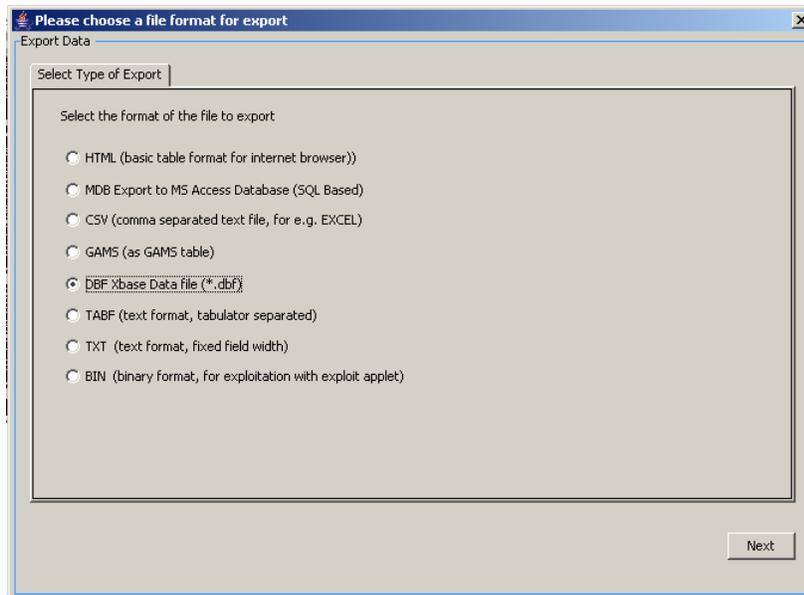


Open the map option dialogue by pressing the map option button “”. Change the settings according to you needs, and then press the “store settings” button in the lower part of the dialogue. Choose a file name and location. You may later used “load settings” to retrieve them again, and apply them to another map.

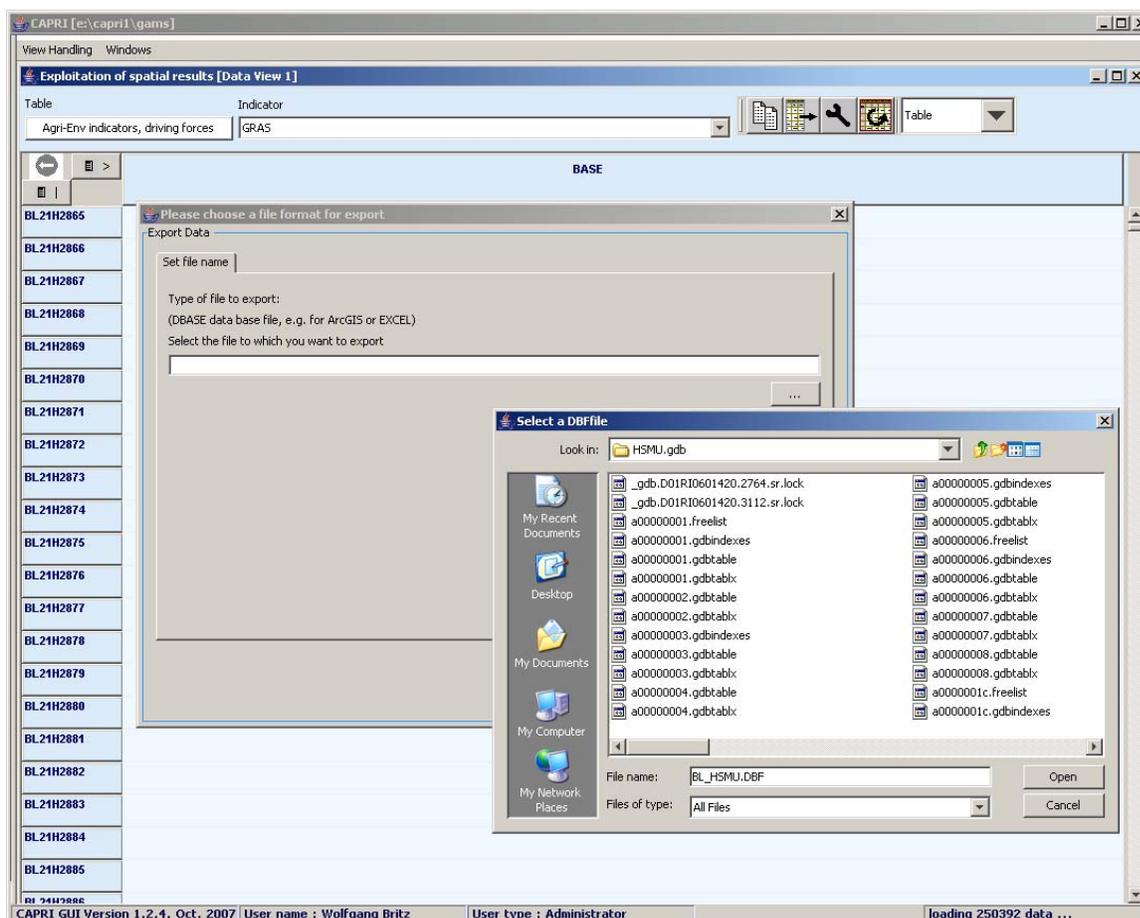


## Exporting the data underlying the map

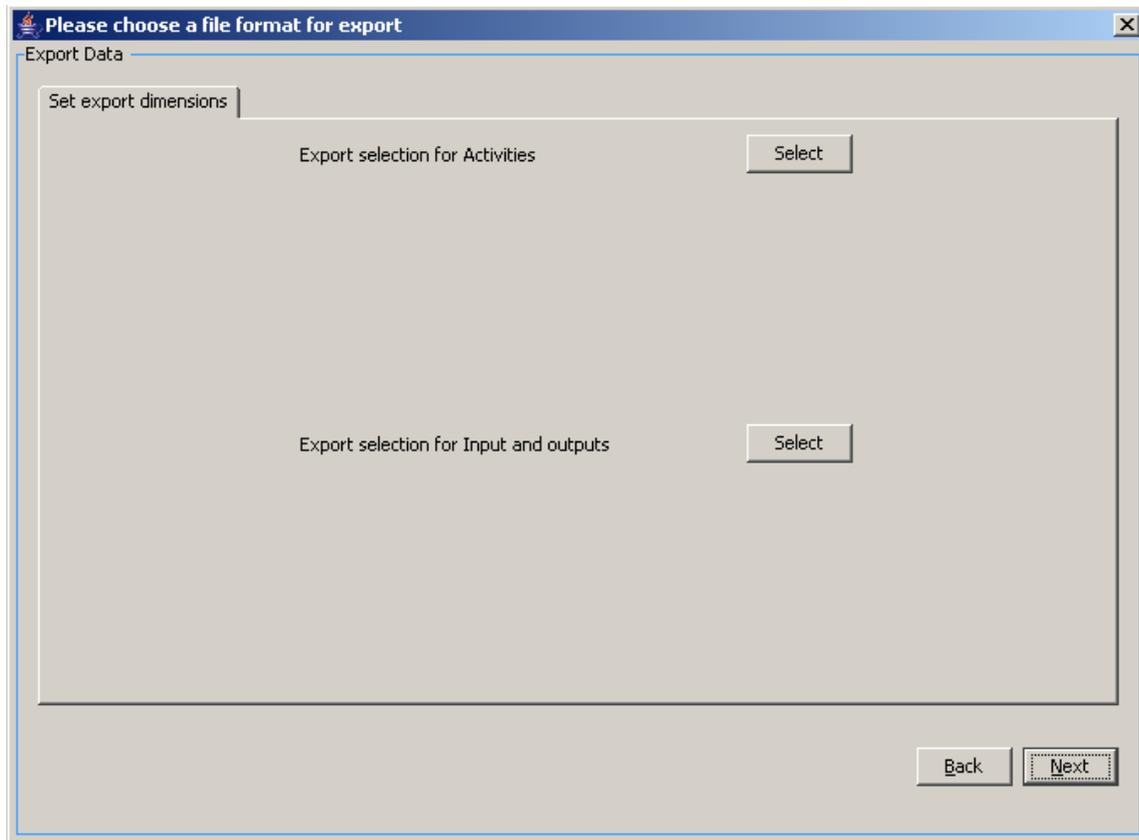
As mentioned above, the mapping viewer is part of the CAPRI exploitation tools which is in its core based on pivot tables. In order to export the data, e.g. to GIS system, the view must first be changed to tables. Afterwards, the  button which will open a file dialog as shown below. For GIS-export, e.g. to ArcGIS, DBF is the recommended format.



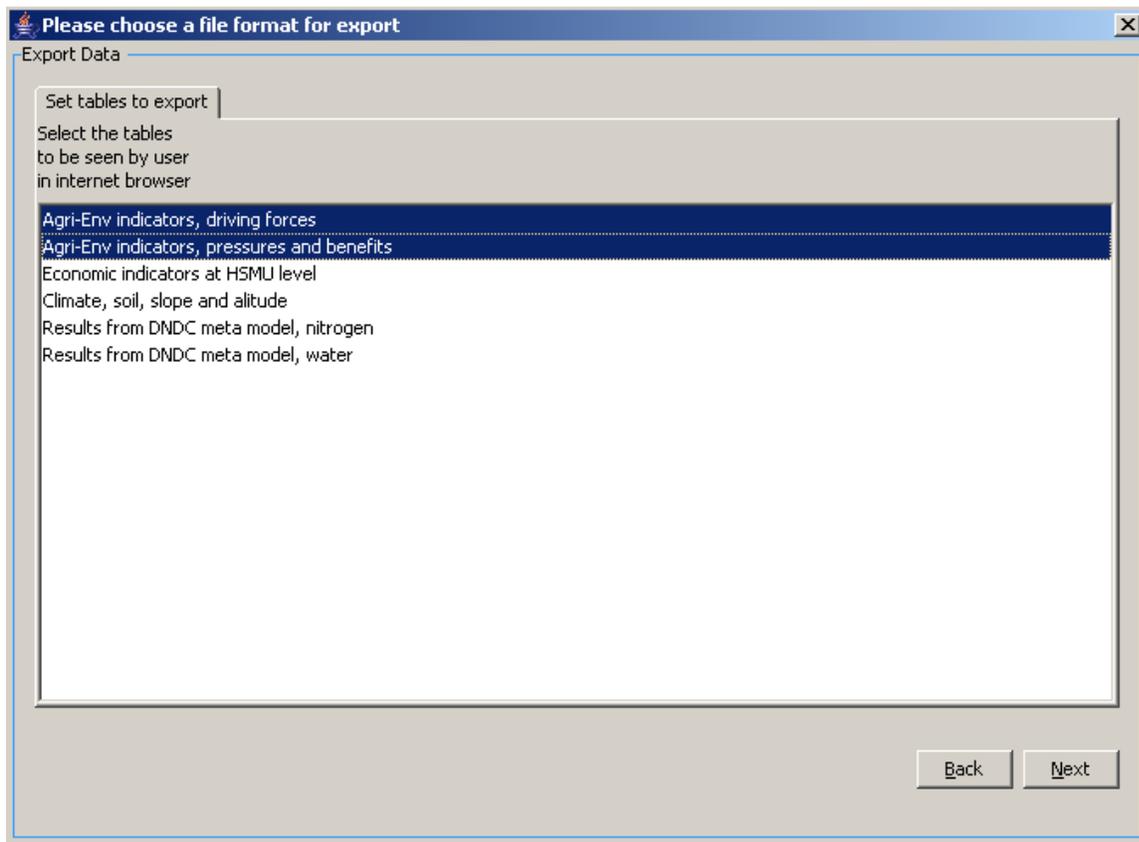
Once next is pressed, the next pane will open a file dialog to choose a file. In the case of export to a Microsoft Access Data Base, the file must exist.



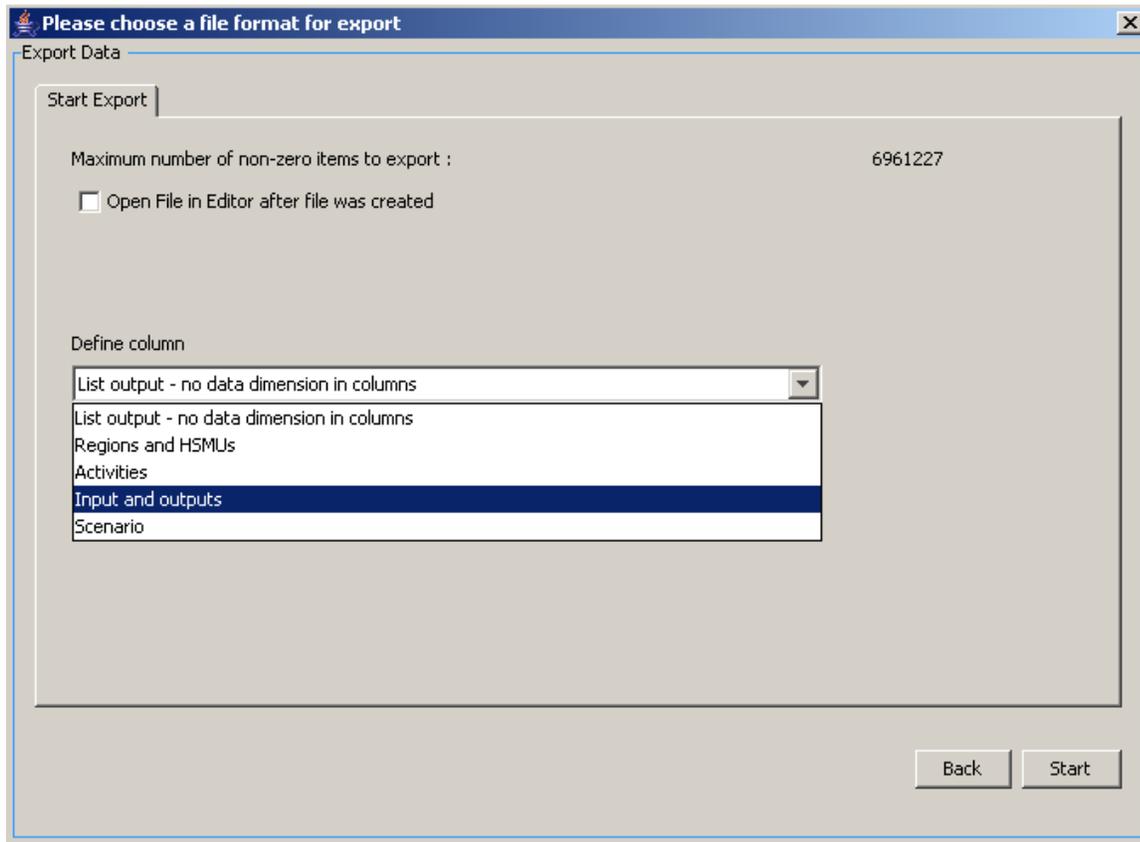
The next pane allows, if wished to open selection lists for the different data dimensions.



Next, the tables for export can be selected.



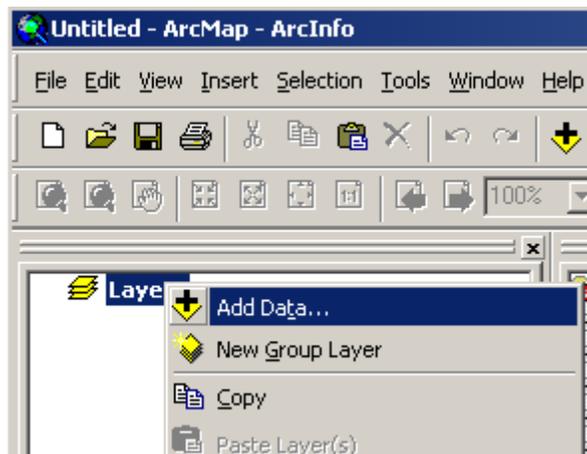
Beware: the pre-defined table structure will be lost, as will the long-texts and units attached to the tables. However, in the case of DBF-export, a second file with that information will be automatically created. If you solely want to export the table you have currently up front, use the “copy to clipboard”  button. The clipboard export will retain the pivoting and further information.



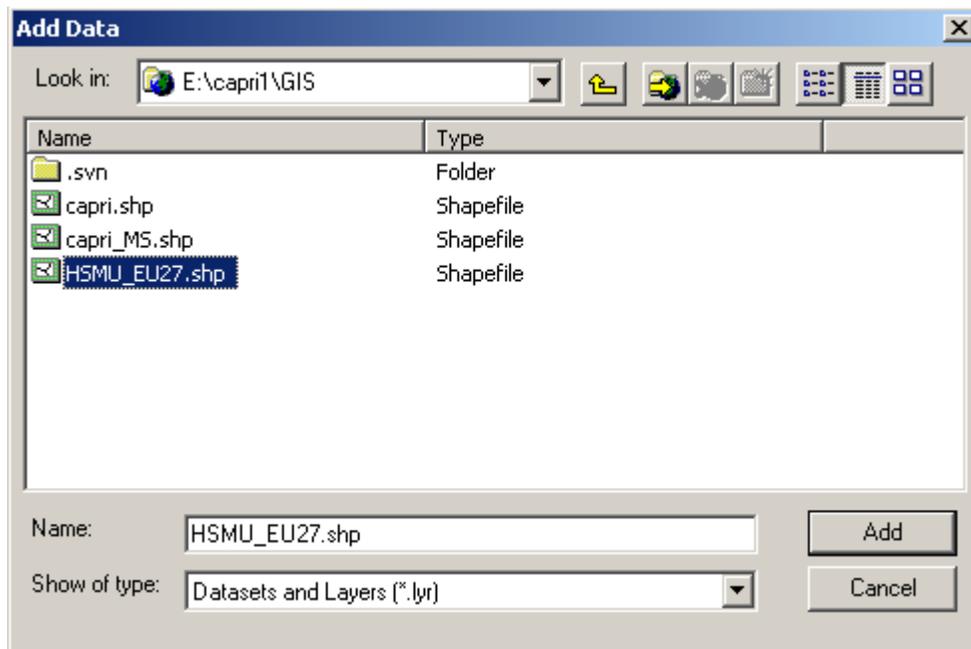
The last pane let's you decide for DBF-export if you want a list, or if you want of the data dimension spanned across the columns. For exporting the HSMU tables, it is recommended to put “Inputs and outputs” in the columns.

If everything has worked well, we should now find two files: one with the data, named as chosen in the file dialog, and a second one with “\_meta” introduced before the file extension.

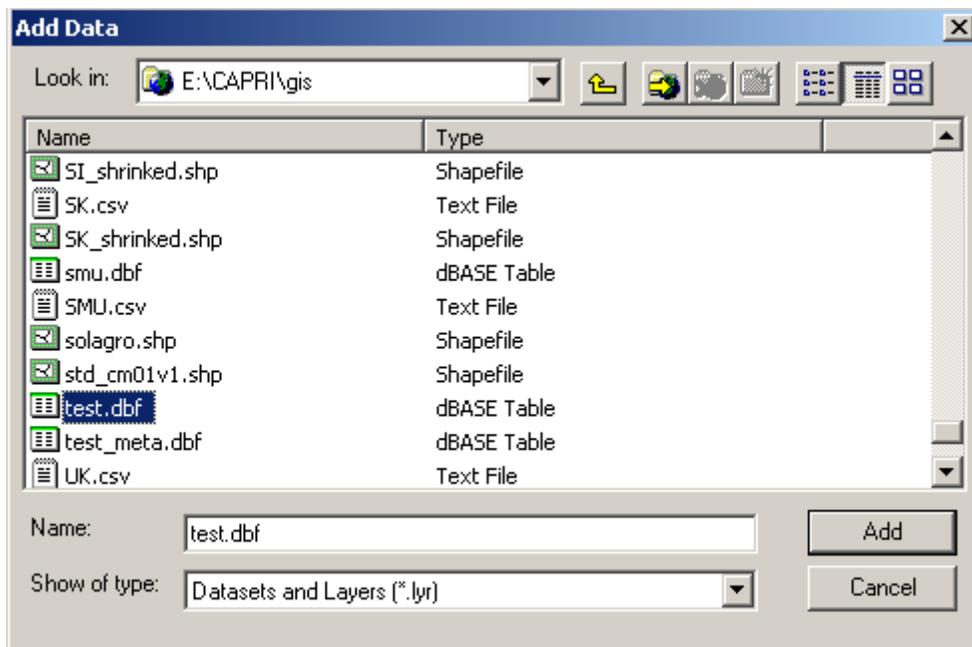
The following section will briefly explain how to the now work with the data in ArcGIS. Under Layers, choose add Data,



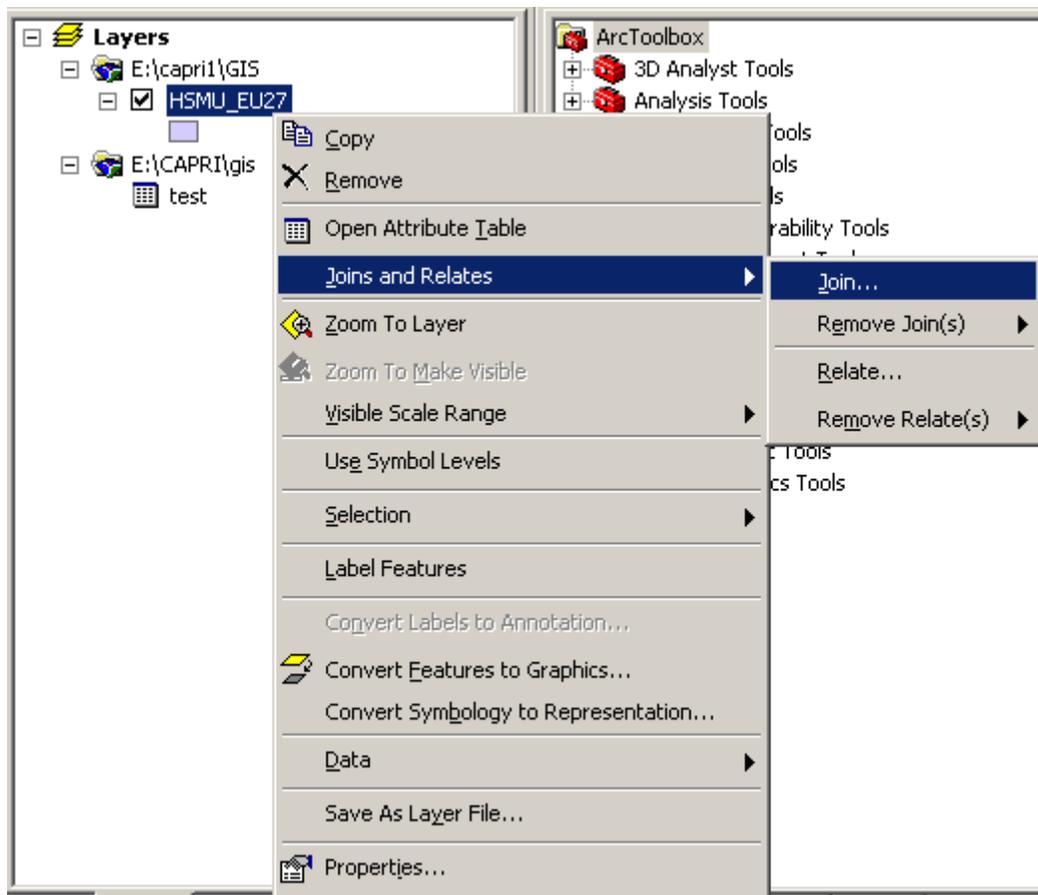
and in the case of the HSMUs, add the “HSMU\_EU27.shp” shapefile.



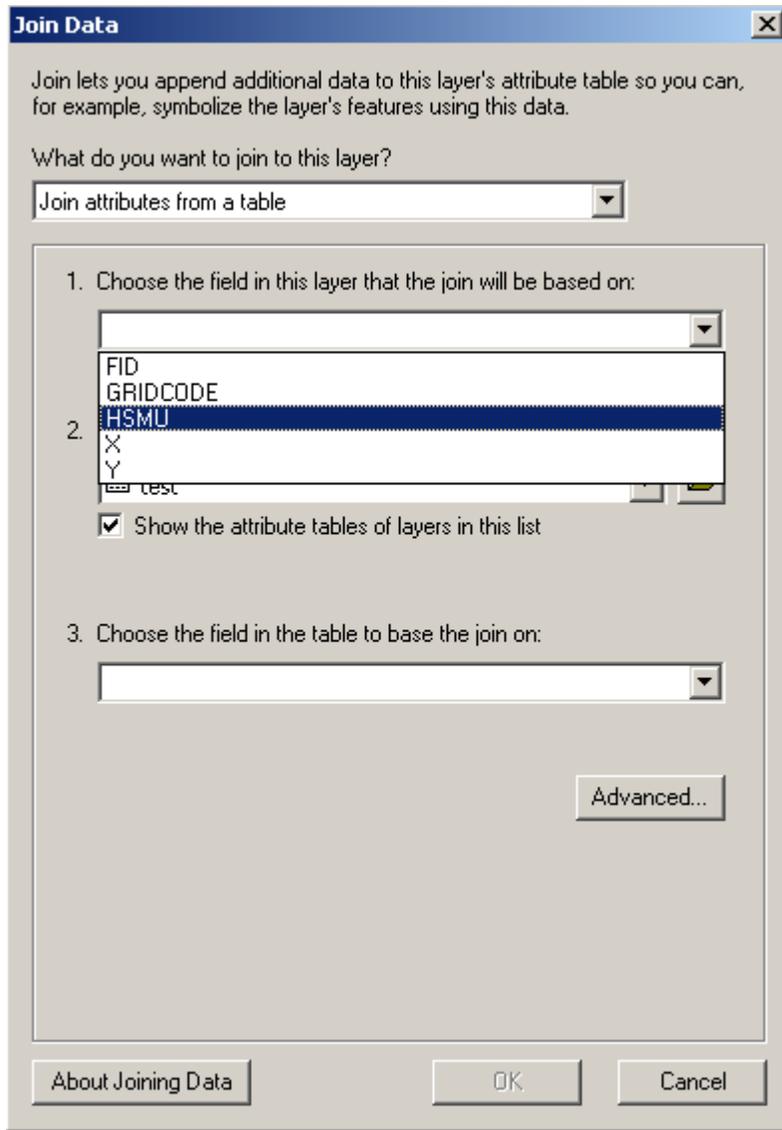
Then, choose add layers again, and add the dbf-file you generate in the step explained above. You may also add the file with the meta data.



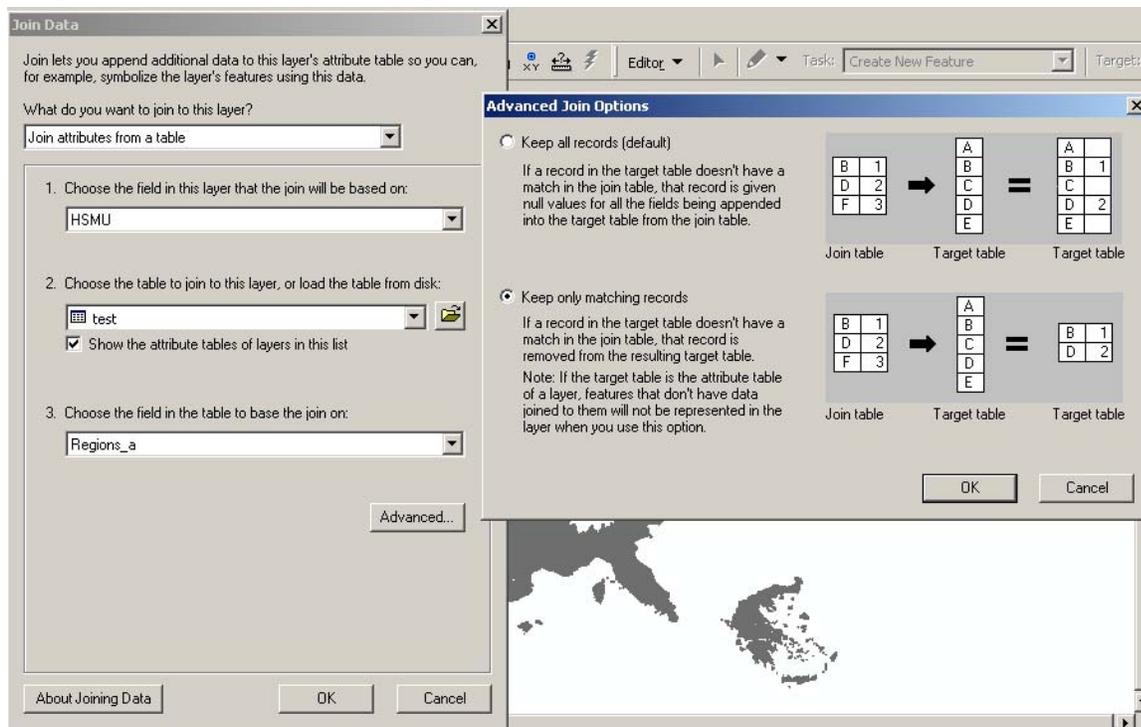
Next, we need to connect the HSMU geometry with the newly loaded data, a process called “joining” in ArcGIS. In the context of “HSMU\_EU27”, choose “Join and Relates”, then “Join ...”.



That will open the join dialogue as shown below.



Make sure that “Join attributes from a table” is set in the first drop down box, and under 1., select HSMU, i.e. the field in the HSMU\_27 geometry where the codes for the HSMU polygons are stored. Use the name of the exported dbf-table under 2., and select the field “Regions\_a” (the field name are restricted to 10 chars) under 3. Then press the button labeled “advanced”, and choose the radiobutton “keep only matching records”. If you are asked to build index, confirm.



If anything has worked well, you should now see the country or countries you had in the original map.

There is a trap, though. If you export several tables, or results for several scenarios, your table will normally have several fields used as a row header (e.g. year, scenario, activity). If that is the case, the join will not work properly as several rows for the same regions will be joined to the very same polygon. Unfortunately, ArcGIS will not warn you about that. You have *first* to execute a definition query in the table, selecting those rows you are later going to draw a map from.

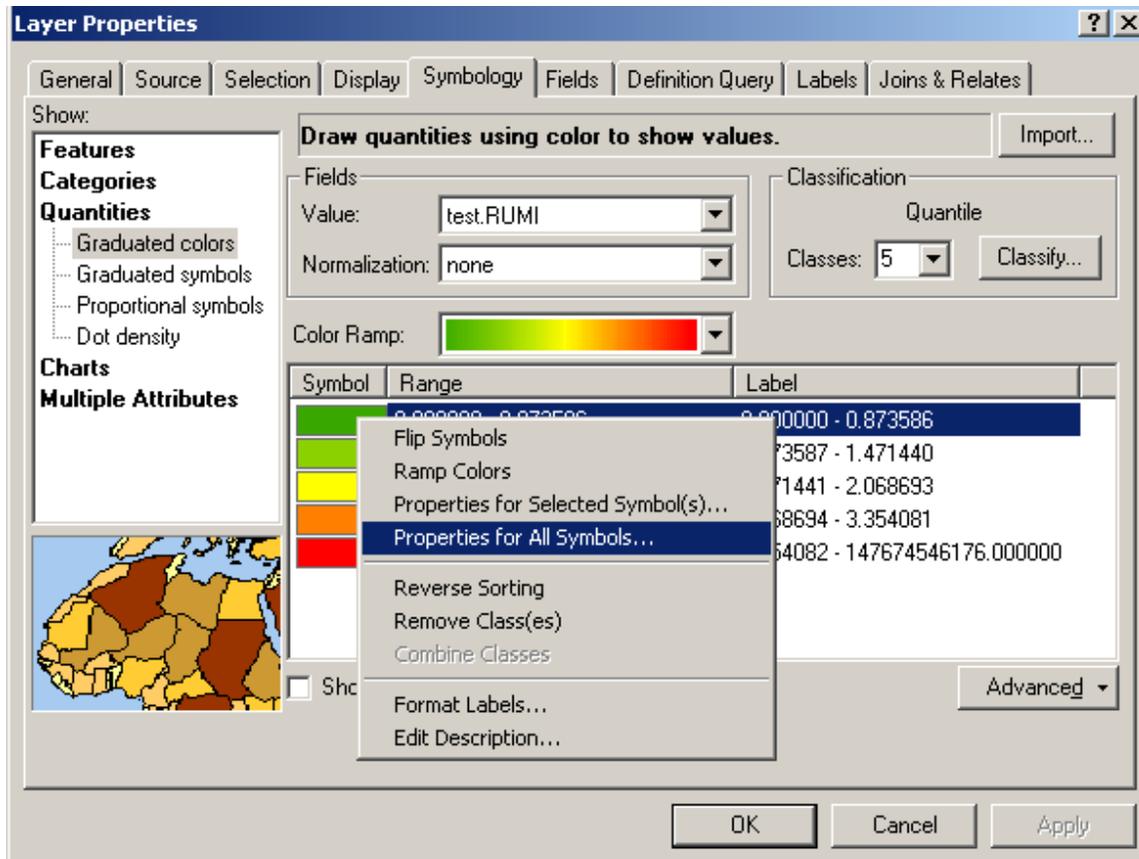
In order to draw now a thematic map, it may be helpful to add the file with the meta data to the map, and with the help of its context menu, open the meta-data table. It will give us the long description and units belonging to the data fields in the exported data table.

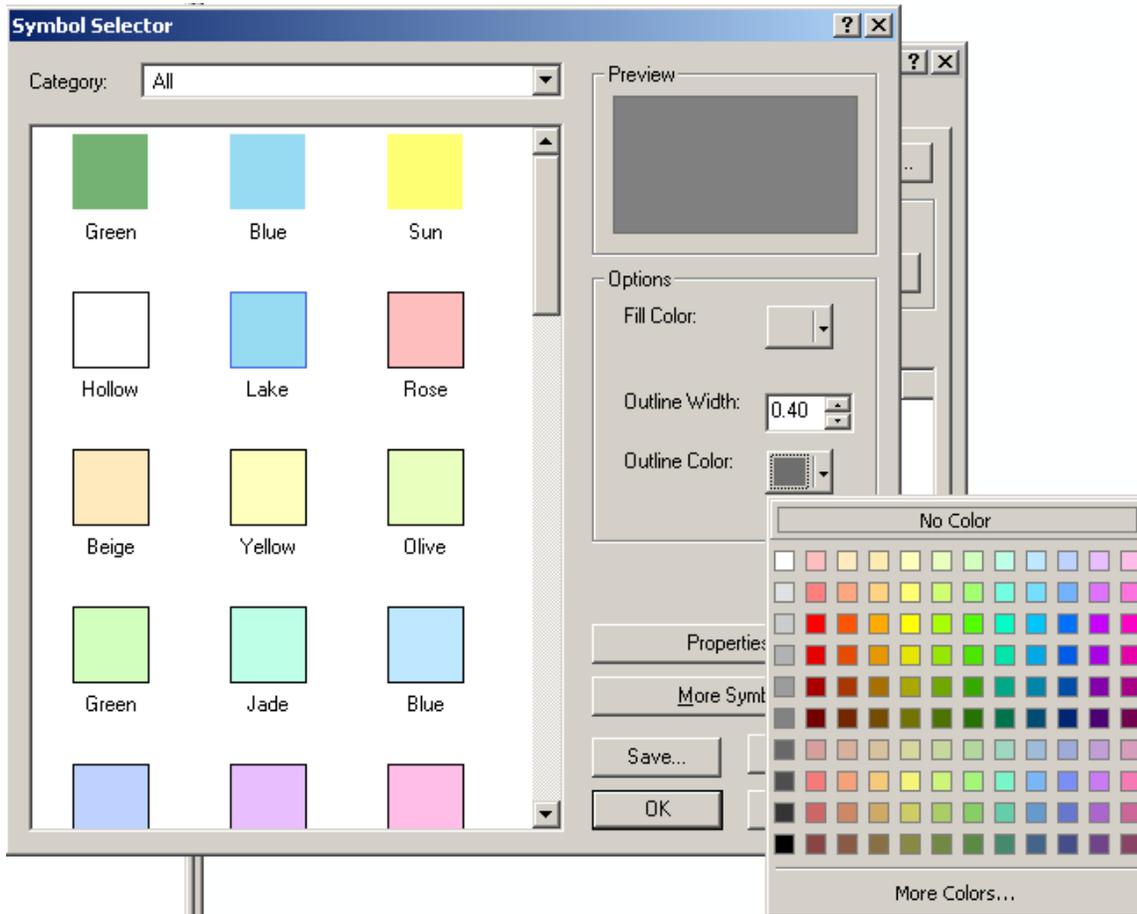
OID	Key	Name	Unit	LongText
0	NMIN	No 5 (a), Mineral Fertilizer Consumption, Nitrogen	kg N/ha	
1	PMIN	No 5 (b), Mineral Fertilizer Consumption, Phosphorous	kg N/ha	
2	NMIN_SWHE	No 5 (c), Mineral Nitrogen Application rate, Soft wheat	kg N/ha	
3	PLAP	No 6, Consumption of Pesticides	Euro /ha	
4	IRR	No 7(a), Irrigation, share	% irrigated	
5	WAT_SURP	No 7(b), Irrigation, abstraction	l/m2	
6	ELEC	No 8 (a), Energy, Electricity	Euro/ha	
7	EGAS	No 8 (b), Energy, Gas	Euro/ha	
8	EFUL	No 8 (c), Energy, Fuels	Euro/ha	
9	LU	No 10 (a), Cropping/Livestock pattern, livestock density	Livestock units / ha UAA	
10	RUMI	No 10 (b) Cropping/Livestock pattern, ruminants density	Livestock units / ha Fodder area	
11	PP	No 10 (c), Cropping/Livestock pattern, non-ruminants density	Livestock units / ha UAA	
12	ARAB	No 10 (d), Cropping/Livestock pattern, arable land density	%	
13	GRAS	No 10 (d), Cropping/Livestock pattern, grass land density	%	
14	PERM	No 10 (d), Cropping/Livestock pattern, permanent crops density(d)	%	
15	INTE	No 12 (a), low-medium-high input farming	Index 0 - 2	
16	H2865	BL21H2865		
17	H2866	BL21H2866		
18	H2867	BL21H2867		
19	H2868	BL21H2868		
20	H2869	BL21H2869		
21	H2870	BL21H2870		
22	H2871	BL21H2871		

Assuming we want to draw now a map with the ruminant stocking density, we find it in row 10 under the key “RUMI”. In order to produce now a map, we have to open the context menu of “HSMU\_EU27”, and choose properties, symbology, and choose “Quantities”. Under values, choose “RUMI”, the name before is the name of the DBF-file.

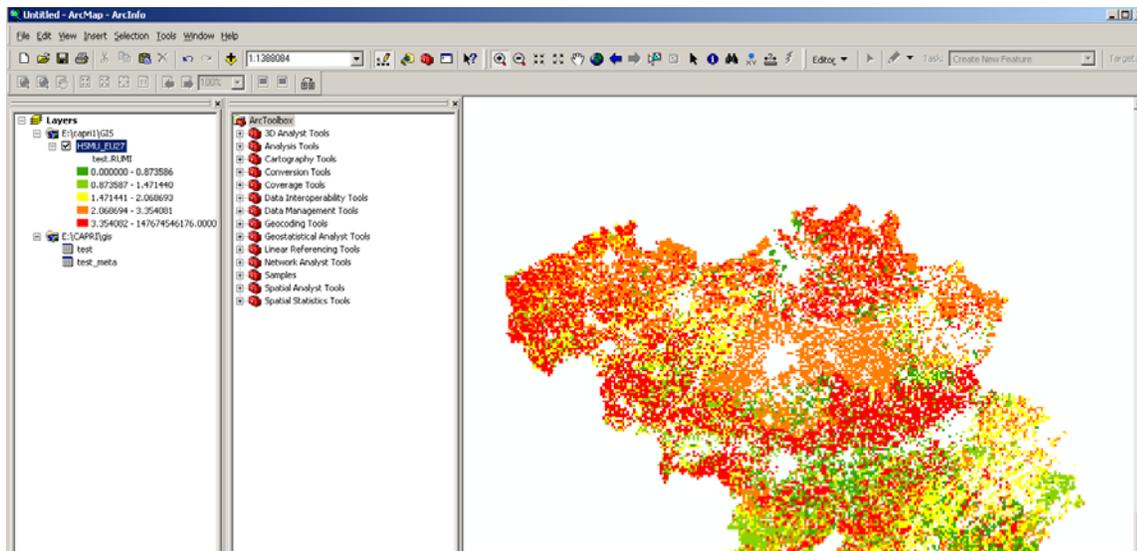
The screenshot shows the 'Layer Properties' dialog box for a layer named 'HSMU\_EU27'. The 'Symbology' tab is selected, and the 'Draw quantities using color to show values' option is checked. A dropdown menu is open, displaying a list of fields: none, HSMU\_EU27.GRIDCODE, HSMU\_EU27.X, HSMU\_EU27.Y, test.OID, test.GRAS, test.LU, test.PLAP, test.ELEC, test.EGAS, test.EFUL, test.NMIN, test.ARAB, test.PERM, test.PP, test.WAT\_SURP, test.RUMI (highlighted), test.NMIN\_SWHE, and test.INTE. The 'Classification' section is set to 'Manual'. The 'Show class names' checkbox is unchecked. A small map preview is visible in the bottom left corner.

Afterwards, under classification, choose your preferred one. As there many small polygons, the outline of the polygons should not be drawn, click on one of the colors, choose “Properties for all symbols ...” and under “Outline color” chose “No Color”.





Afterwards, if anything went well, you should see your map.



## ***What are the HSMUs and what information is available?***

The HSMU are the so-called Homogenous Soil Mapping Units. Each HSMU contains one or several 1x1 km grid cells, not necessarily adjacent, and are defined so that these are more or less homogenous regarding climate, soil, slope, CORINE land cover class and NUTS II region. There are about 110.000 HSMUs for EU15. The spatial downscaling introduced in CAPRI-Dynaspat provides the following information per HSMU:

- Cropping shares and animal stocking densities
- Yields
- Economic indicators per crop and animal, and in relation to UAA
- Fertilizer application rates
- Environmental indicators

## ***How to visualize the HSMU information***

Given the 1x1 grid resolution, the most obvious way to look at the information is to produce maps with the CAPRI GUI. There is a co-ordinate set available which is called “HSMU.zip” which comprises the geometry for about 1.8 Mio Polygon which represent the HSMUs.

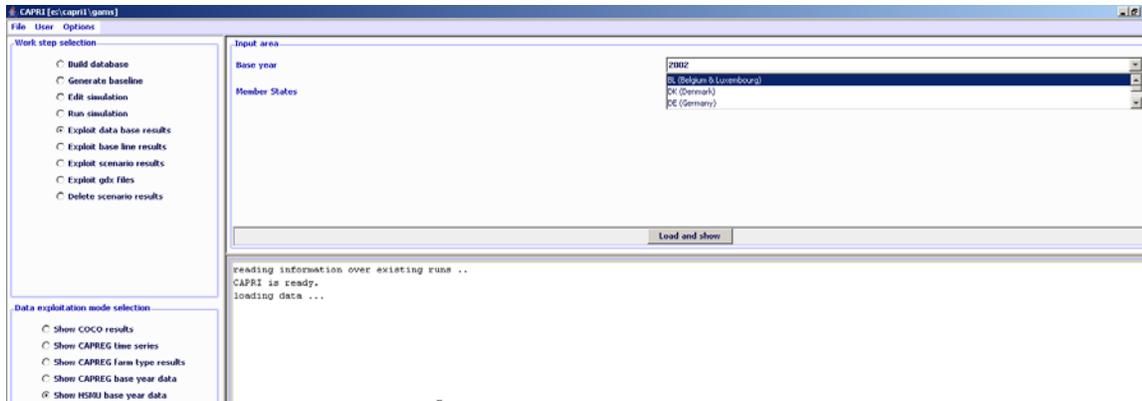
There are four options to view HSMU data:

1. Loading data for one or several Member States for the base year (dis-aggregated information from the NUTS II CAPRI data base).
2. Loading data for one or several Member States for the base year and the baseline, the latter representing dis-aggregated data from NUTS II results of the baseline calibration.
3. Loading data for one or more scenarios for a given year.
4. Loading data manually

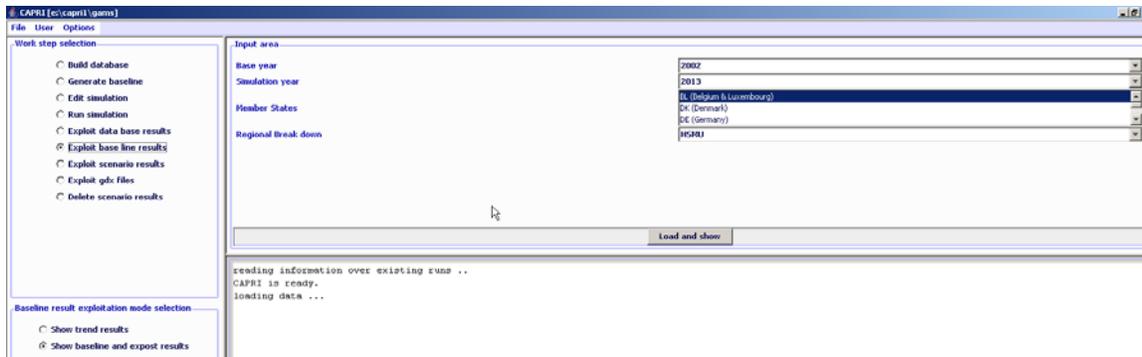
## ***Loading data based on the GUI presets***

As for the results at NUTS II level, there are three pre-defined exploitation possibilities included in the CAPRI GUI:

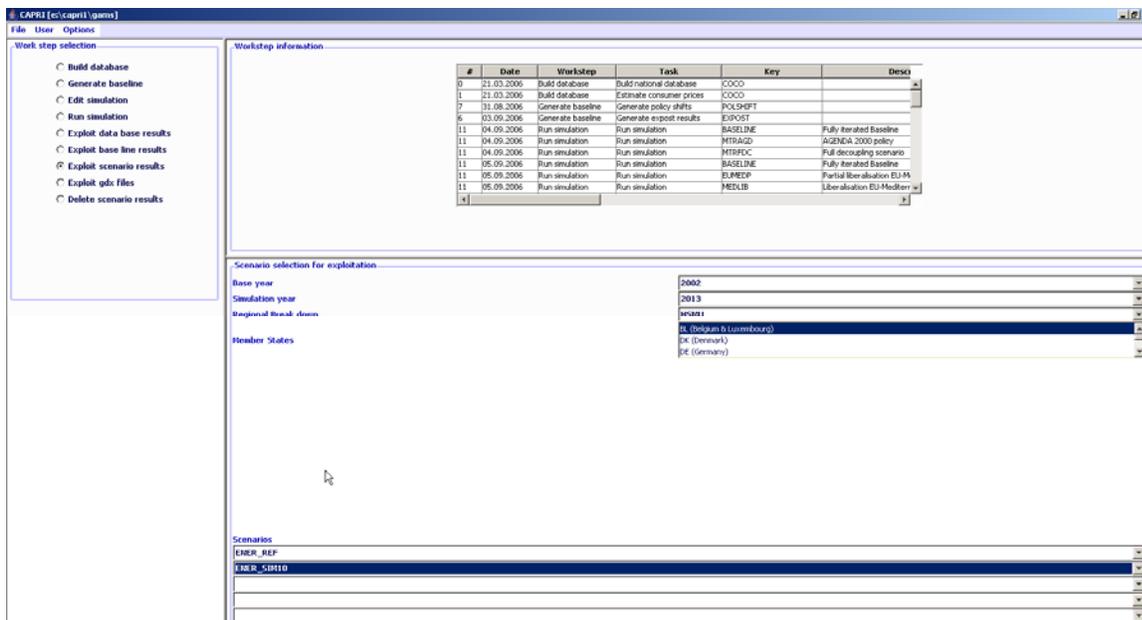
1. Viewing the results for the base year. Given the tremendous number of HSMUs, the user can select for which Member States to the load the information.



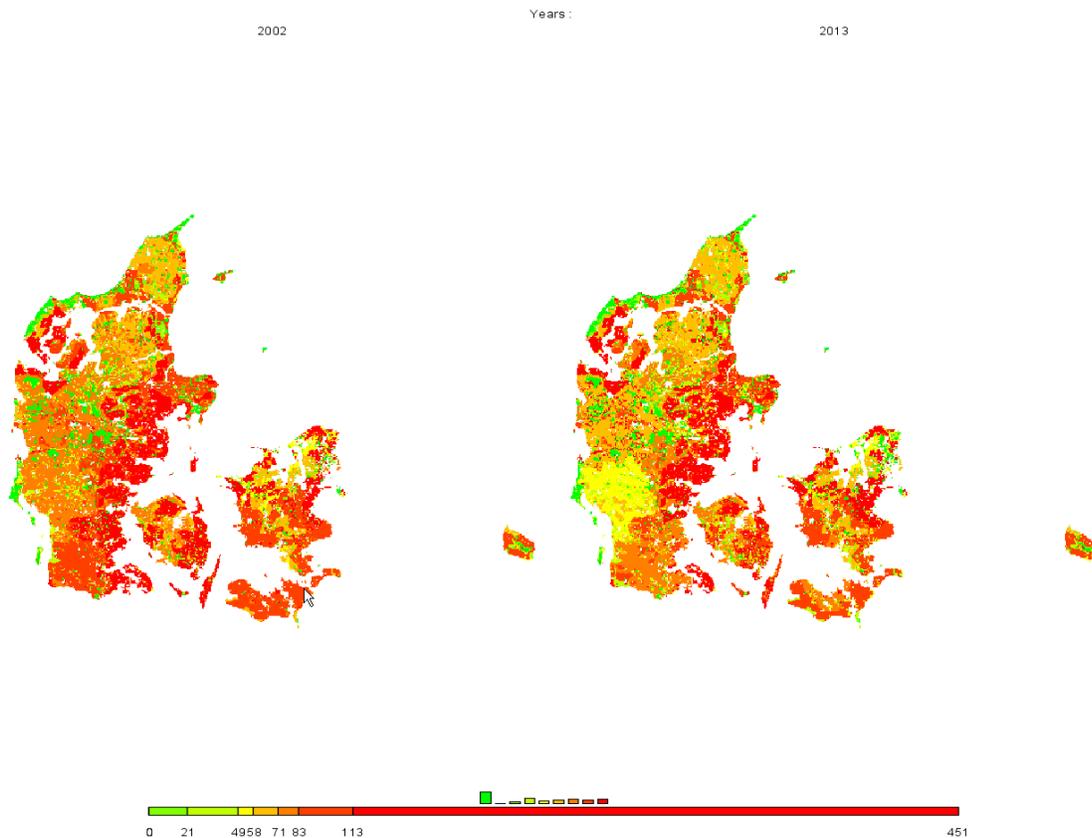
2. Comparing the results for the base with the baseline (projection results), comparison between two points in time. Again, the user can select the Member States.



3. Comparing results from different scenarios (comparisons for one point in time, but changes in drivers/assumptions relevant for the CAPRI economic model).



Once the loading is accomplished, the right hand side of the GUI is filled with a tabular view of the results, which can be turned easily into maps (other forms of exploitation as graphs are less suited given the large number of observations). Details how to work with the exploitation tools are found in a separate document. The screen-shot below shows results for Denmark as an example.



In order to ease exploitation of the results, pre-defined tables are set up, currently broken down into five categories:

1. Agri-environmental indicators, driving forces (mineral fertilizer consumption, consumption of pesticides, irrigation shares, energy consumption, livestock densities, shares of arable / grass land or permanent crops)
2. Agri-environmental indicators, pressures and benefits (Gross nitrogen and phosphorous balance, green house gas emissions, High Nature Value Farm land indicator)
3. Economic indicators at HSMU level (market revenues, variable production costs, income)
4. Climate, soil, slope and altitude
5. Results from the DNDC meat model (gas losses for different nitrogen compartments, mineralization, leaching)

The tables on agri-environmental indicators (driving forces, pressures and benefits) are set-up as close as possible according to the official EU Guidelines for Rural Development indicators.

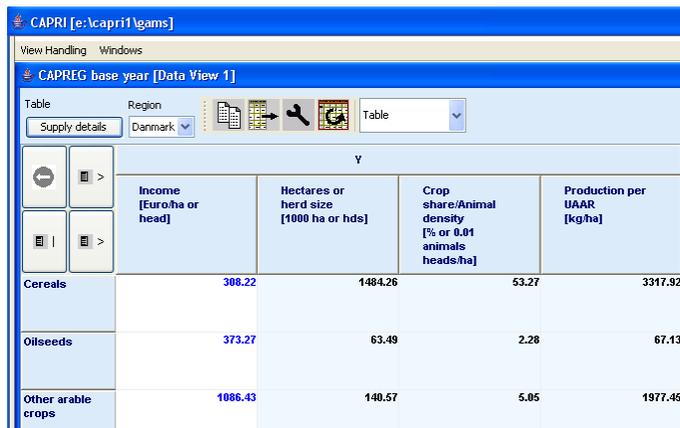
The screenshot displays the CAPRI software interface. On the left, there is a 'Work step selection' panel with options like 'Build database', 'Generate baseline', 'Edit simulation', 'Run simulation', 'Exploit data base results', 'Exploit base line results', 'Exploit scenario results', 'Exploit gdx files', and 'Delete scenario results'. Below it is a 'Baseline result exploitation mode selection' panel with options 'Show trend results' and 'Show baseline and exploit results'. The main window shows a table titled 'Scenario exploitation [Data View 1]' with columns for 'Activity', 'Items', and 'Table'. The table has two columns for the years 2002 and 2013. The data is as follows:

Activity	Items	Table	2002	2013
	No S (a), Mineral Fertilizer Consumption, Nitrogen kg N/ha	Agri-Env indicators, driving Forces		
H22500				
H22509				
H22510				
H22511				
H22512				
H22513				
H22514				
H22515				
H22516				
H22517				
H22518				
H22602			51.77	51.84
H22603			104.82	118.87
H22604			103.53	117.63
H22605			64.34	69.89
H22606			65.15	67.51
H22607			128.25	135.90
H22608			132.88	146.79
H22609			78.41	71.88
H22610			69.56	70.83
H22611			132.88	138.82
H22612			123.17	129.35
H22613			56.99	53.48
H22614			55.87	52.14
H22615			52.52	49.16
H22616			63.58	67.83
H22617			55.88	57.88
H22618			48.33	62.82
H22619			38.84	36.16
H22620				
H22621				
H22622				
H22623				
H22624			51.87	38.52
H22625			69.95	72.25
H22626			126.26	131.88
H22627			62.88	62.87
H22628			66.58	65.29

## Examples

### ***Drawing a map showing the nitrate surplus for EU27 at regional level in the base year***

Firstly, we need to select “Exploit data base results” in the work step selection panel, and then choose the radio button “Show CAPREG base year data”. Then, in the Member States drop-down list, right click the mouse, and select “EU27”. The “Load and show button will then load the results.



The screenshot shows the CAPRI software interface. The title bar reads "CAPRI [e:\capri1\gams]". Below the title bar, there are menu options "View Handling" and "Windows". The main window title is "CAPREG base year [Data View 1]". The interface includes a "Table" section with a "Region" dropdown set to "Denmark" and a "Table" dropdown. Below this, there are navigation buttons (back, forward, search, etc.). The main data table is titled "Y" and has the following columns: "Income [Euro/ha or head]", "Hectares or herd size [1000 ha or hds]", "Crop share/Animal density [% or 0.01 animals heads/ha]", and "Production per UAAR [kg/ha]". The data rows are:

	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Crop share/Animal density [% or 0.01 animals heads/ha]	Production per UAAR [kg/ha]
Cereals	308.22	1484.26	53.27	3317.92
Oilseeds	373.27	63.49	2.28	67.13
Other arable crops	1086.43	140.57	5.05	1977.45

Next, select a different selection of tables by pressing on the button below “Table”, which currently shows the topic “Supply details”. In the drop-down list, go-to “Environment” and select “Nutrient balances, mapping view”.

CAPRI [e:\capri1\gams]

View Handling Windows

CAPREG base year [Data View 1]

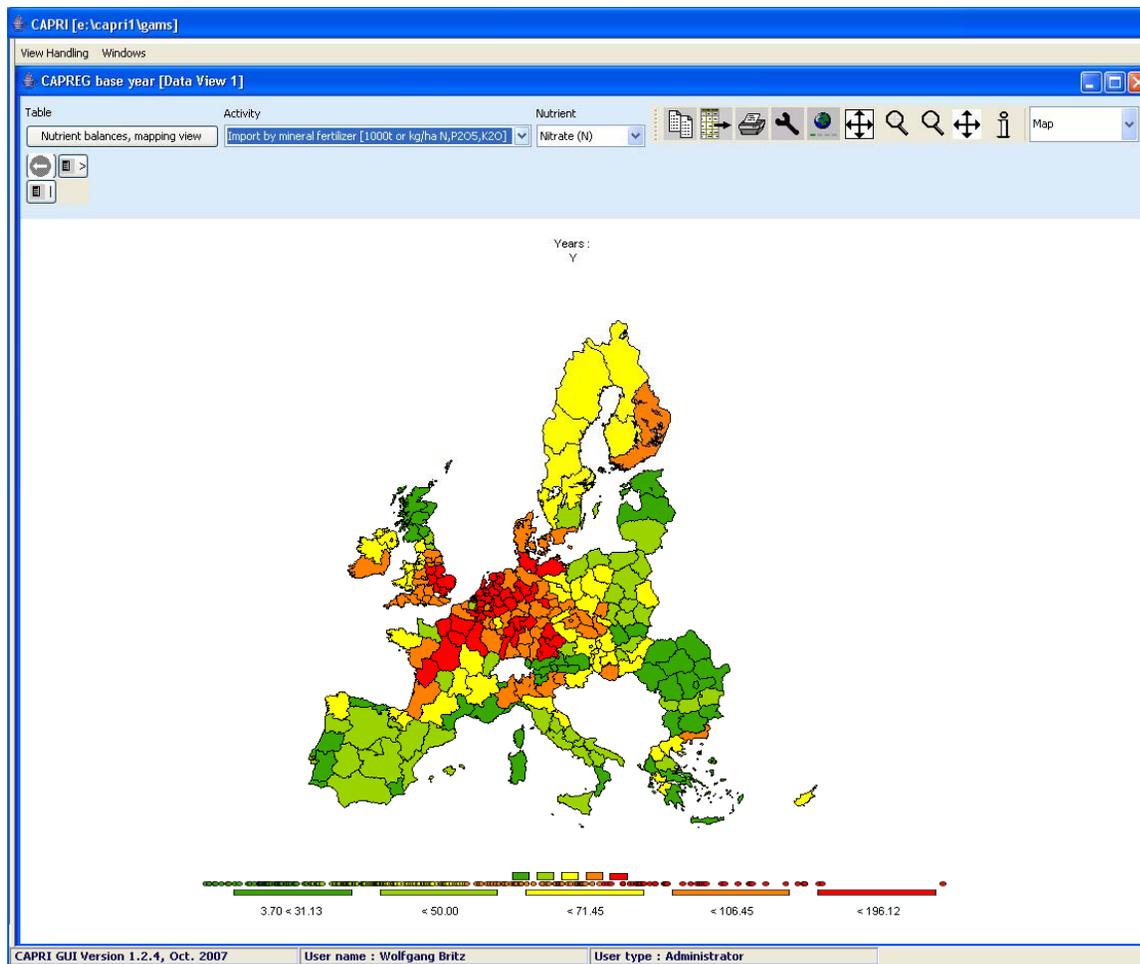
Table Region

Welfare ▶ mark  
 Markets ▶  
 Prices ▶  
 Farm ▶  
 HSMU ▶  
 DNDC ▶  
 Environment ▶  
 Multi-Functionality ▶  
 Energy ▶  
 No table

Y			
Income [euro/ha or ad]	Hectares or herd size [1000 ha or hds]	Crop share/Animal density [% or 0.01 animals heads/ha]	Production per UAAR [kg/ha]
		53.27	3317.92
		2.28	67.13
		5.05	1977.45

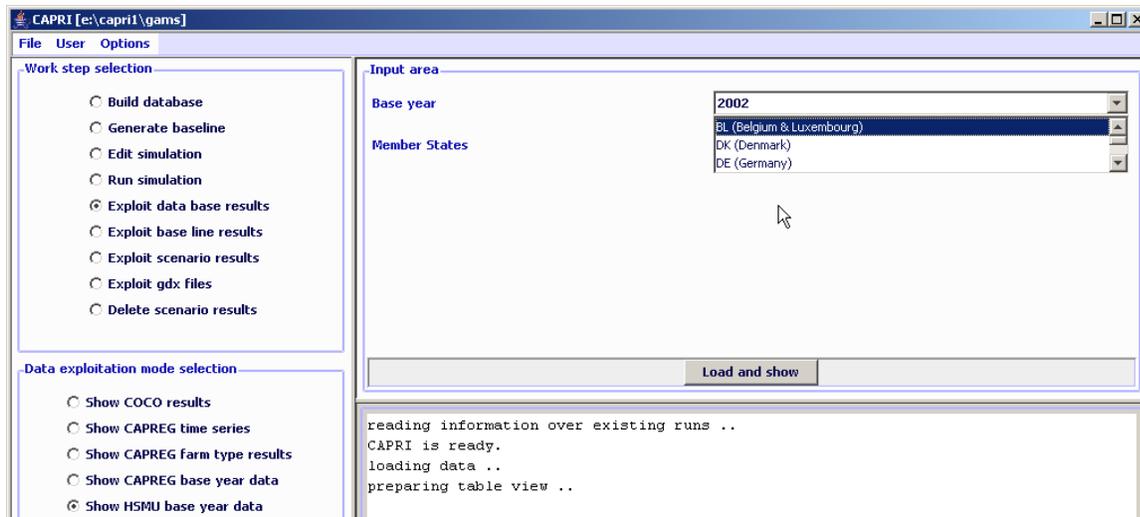
Manure output per animal  
 Environmental indicators per activity  
 Nutrient balances  
 Nutrient balances - soil details  
 Nutrient balances - gas losses  
 Nutrient balances, compare Member States  
 Nutrient balances, mapping view  
 Methane emissions  
 N2O emissions  
 Energy and Ressource consumption

The following map should appear. You may select now different elements of the balance, by using the drop-down boxe under “activity” or change the nutrient, by using the drop-down box under “nutrient”.

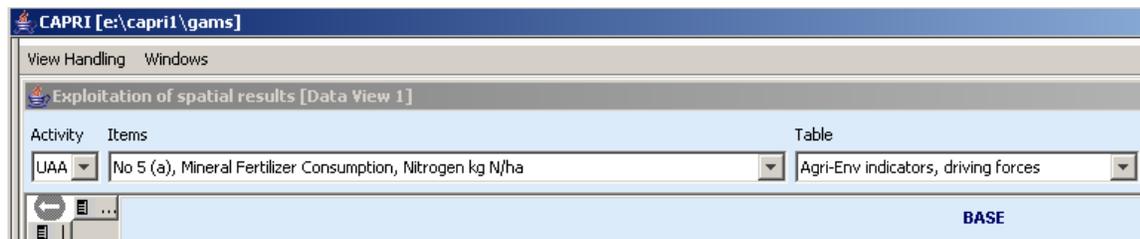


***Drawing a map of the High Nature Value Farmland indicator for Belgium for the base year***

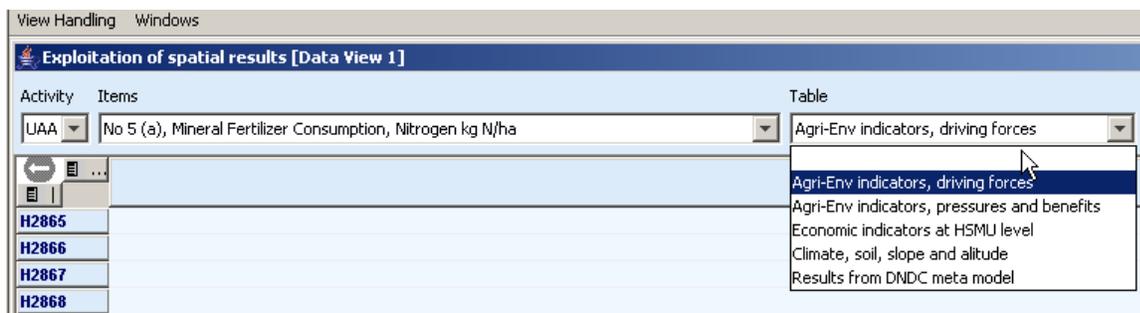
Firstly, we need to select “Exploit data base results” in the work step selection panel, and then choose the radio button “Show HSMU base year data”. Then, in the Member States drop-down list, Belgium must be selected, as shown below.



Pressing the load and show button will then bring up the first table links to the HSMU results as shown below.



Now, according to the guidelines for RD indicators, HNV is to be found under “Pressures and benefits”, which is stored under another table. So opening the drop down list for the tables allows us to select the correct table.



Next, we need to find the correct item, to do so, the drop-down lists for the items must be selected, and the indicator 23 selected, as seen below.



**Map option dialogue**

Color table: Start color Mid color End color Green yellow red  
 Set value for middle color 2,96  
 Treat zeros as missing values, exclude from classification and don't draw  
 Use area weights for classification

Classification method: Equal interval  
 Number of classes: 10  
 Number of regions with small values to remove from class definition: 0  
 Number of regions with large values to remove from class definition: 0

Legend: Continuous bar, linear scale

#	label	class limit	% of obs	color
1	0.00 < 0.99	0.998	28.034	Green
2	< 1.98	1.975	9.972	Light Green
3	< 2.96	2.963	18.063	Yellow-Green
4	< 3.95	3.951	7.123	Yellow
5	< 4.94	4.938	9.744	Light Yellow
6	< 5.93	5.926	6.781	Orange
7	< 6.91	6.914	7.066	Light Orange
8	< 7.90	7.901	4.9	Red-Orange
9	< 8.89	8.889	5.812	Red
10	< 9.88	9.877	2.507	Dark Red

Cumulative distribution graph: Frequency groups: 100  Draw mean and +/- 1 std. dev.

Show small circles, showing distribution of regions n: 1755.0  
 Show rectangle representing distribution of classes Min: 0.0  
 Draw in high quality Mean: 3.227795  
 Shrink polygons according to share of UAA Median: 2.6468632  
 Std. Dev: 2.772104

Draw outline in same color

Dimension shown in columns of result window for current region: Scenario  
 Dimension shown in rows of result window for current region: Activity  
 Title on top of map: Standard title

ok

**Customize view**

Font: Arial | 11 | Italic

Fraction digits and decimal separator: 2 | .

Column width: 67  
 Row width: 62

Separator between merged data dms: 6

Use default pivoting for tables  
 Hide empty rows  
 Hide empty columns

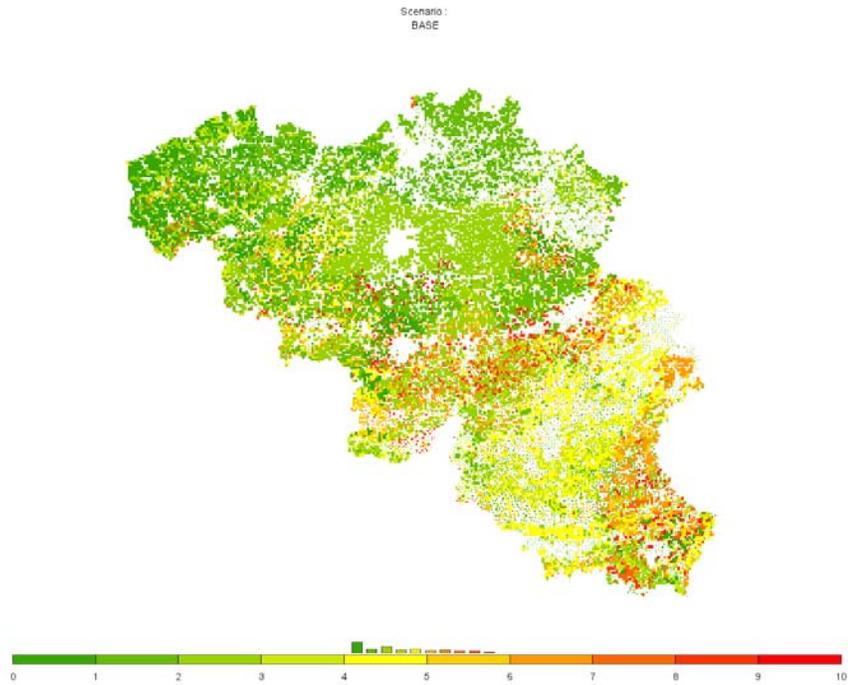
Show only selected items

Long texts only

Comparison output: Only values  
 Data dimension used for comparisons: HMSUs  
 Element used for comparisons: BE21HC965

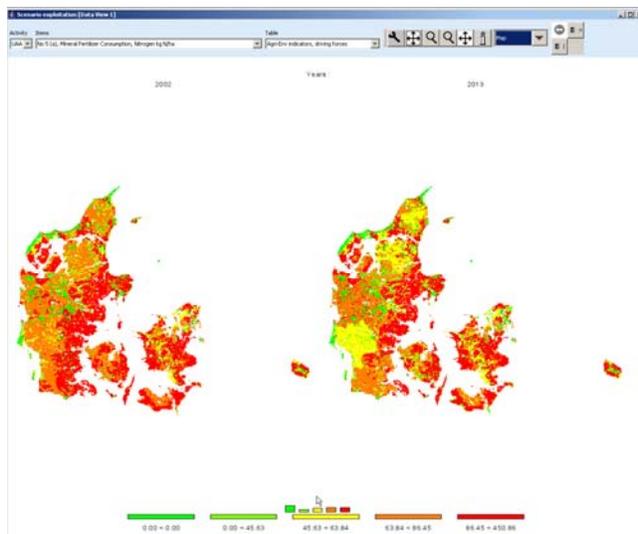
ok

That should give the following map which then can be exported to other application via the clipboard  or can be send to the printer :



## Drawing a map showing changes between the base year and the ex-ante results from the baseline

When scenarios or different points are compared with each other, it is often useful to draw maps which show relative or absolute changes. The following map is the typical starting point when the baseline is analyzed: two maps with identical class definitions, one for the base and one for projection year.

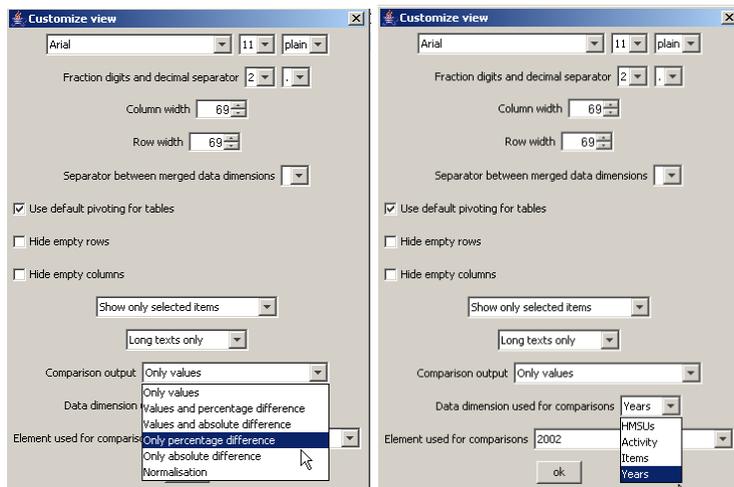


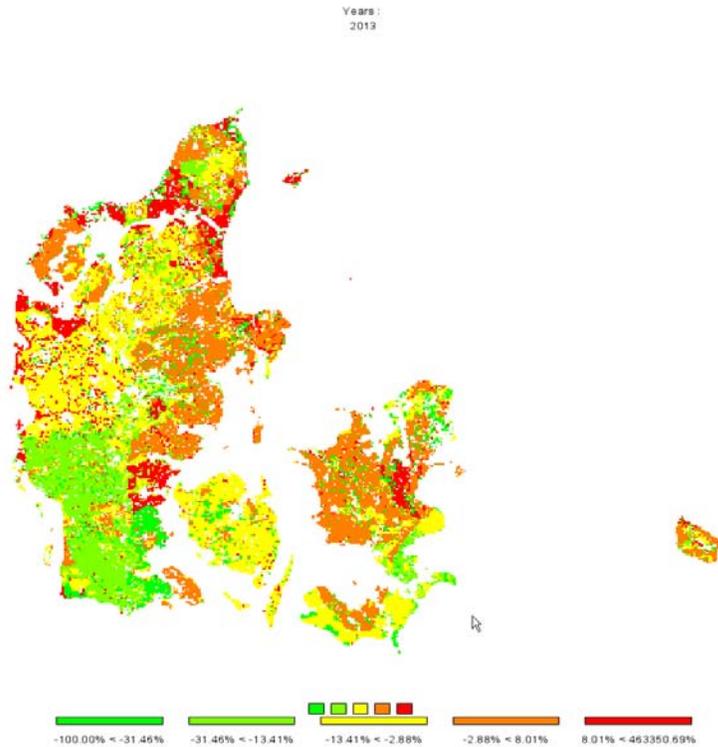
In order to draw a map with changes, we must first get rid of the base year by de-selected the first map.

This is done by using the “column selection” button  which is found in upper right corner of the window. When the button is pressed, a dialog opens, and one can select with the mouse the projection year, only. Afterwards, the left map will not longer be present.



Next, we need to change what is shown in the map to relative changes to the base year. That can be accomplished by using the tool dialogue (press  button). In the tool dialogue, select “only percentage differences” in the drop-down box labeled “comparison output”, and then put the “data dimension used for comparisons” to “Years”. The “Element used for comparisons” should be “2002”. After pressing o.k., the map will change as shown.





Now, there are two things we would most probably like to change: the number of digits shown in the legend, and getting rid of very large values shown in the legend. The number of digits can be changed again with the “tool” dialogue by changing the fraction digits. The large number can be excluded from the classification by increasing the “Number of regions with larger numbers to exclude from class definition”, in the example below the number had been set to 20.

Map option dialogue

Color table: Green yellow red

Classification method: Quantile

Number of classes: 5

Number of regions with small values to remove from class definition: 0

Number of regions with large values to remove from class definition: 20

Separate rectangles:

#	label	class limit	color
1	-100% < -32%	-32.383	Green
2	-32% < -14%	-13.508	Light Green
3	-14% < -4%	-4.063	Yellow
4	-4% < 7%	7.302	Orange
5	7% < 163%	162.746	Red

62

-100    -32    -14    -4    7    31    163

Show small circles, showing distribution of regions

Show rectangle representing distribution of classes

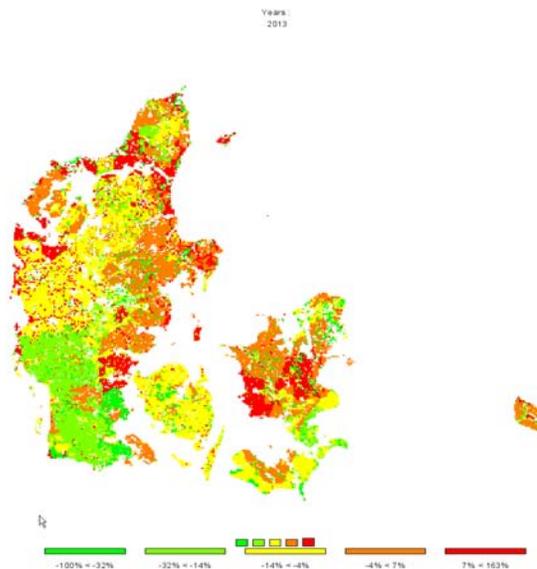
Draw no outline:

Set value for color change from green yellow red

Dimension shown in columns of result window for current region: Years

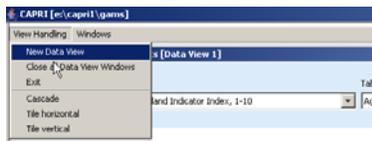
Dimension shown in rows of result window for current region: Activity

ok



## Drawing a map with the base year results next to one showing changes from the base year to the baseline results

There are two ways to draw different maps. If more than one column is selected in the underlying table view, several maps with identical coloring and scaling will be drawn as shown in the sample above. That is not advisable – in our example we need to different scales, one appropriate for the absolute values and one appropriate for changes. In order to do so choose “View Handling”, “New Data View” and then “Tile vertical”.

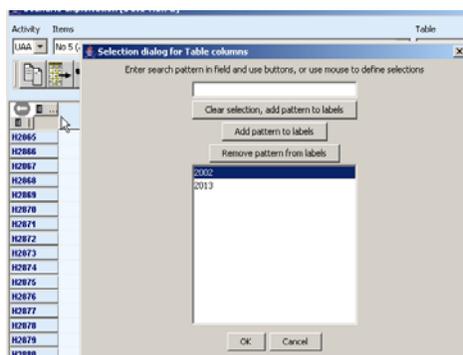


We will then see something as shown below.

The screenshot shows two side-by-side data views in the CAPRI software. Both views display a table with columns for the years 2002 and 2013. The rows represent various activities (H2005 to H2009). The data values are identical in both views, representing the baseline results for the years 2002 and 2013.

Activity	2002	2013
H2005	154.89	178.40
H2006	165.89	209.40
H2007	122.77	223.82
H2008	119.37	207.81
H2009	95.68	116.22
H2070	90.81	113.86
H2071	86.37	167.78
H2072	77.68	128.46
H2073	155.88	182.24
H2074	75.71	167.21
H2075	28.18	34.82
H2076	28.90	68.61
H2077	29.99	106.15
H2078	30.67	66.87
H2079	47.64	59.47
H2080	46.99	46.99
H2081	53.06	281.88
H2082	53.06	281.88

Now, in the left hand side, only the results for the base year should be shown. That can be accomplished by deselecting the column for “2013” – use the column selection button to do so.



The very same trick should be applied for the other view, only deselecting the results for “2002”. Now we see something as below:

2002		2013	
H2865	154.99	H2865	178.40
H2866	195.89	H2866	289.48
H2867	122.77	H2867	229.82
H2868	119.37	H2868	287.81
H2869	95.88	H2869	116.22
H2870	98.81	H2870	113.86
H2871	86.37	H2871	167.78
H2872	77.68	H2872	128.46

Now, for the map with the results for 2013, we should switch to percentage different to the “2002” by opening the “tool” dialog . There, under “Comparison output” choose ”Only percentage difference”, Select “Years” under “Data dimension used for comparisons” and ensure that “2002” is selected in “Element used for comparisons”.

Customize view

Font: Arial, Size: 11, Style: plain

Fraction digits and decimal separator: 2, .

Column width: 67

Row width: 62

Separator between merged data dimensions: [dropdown]

Use default pivoting for tables

Hide empty rows

Hide empty columns

Show only selected items: [dropdown]

Long texts only: [dropdown]

Comparison output: Only percentage difference

Data dimension used for comparisons: Years

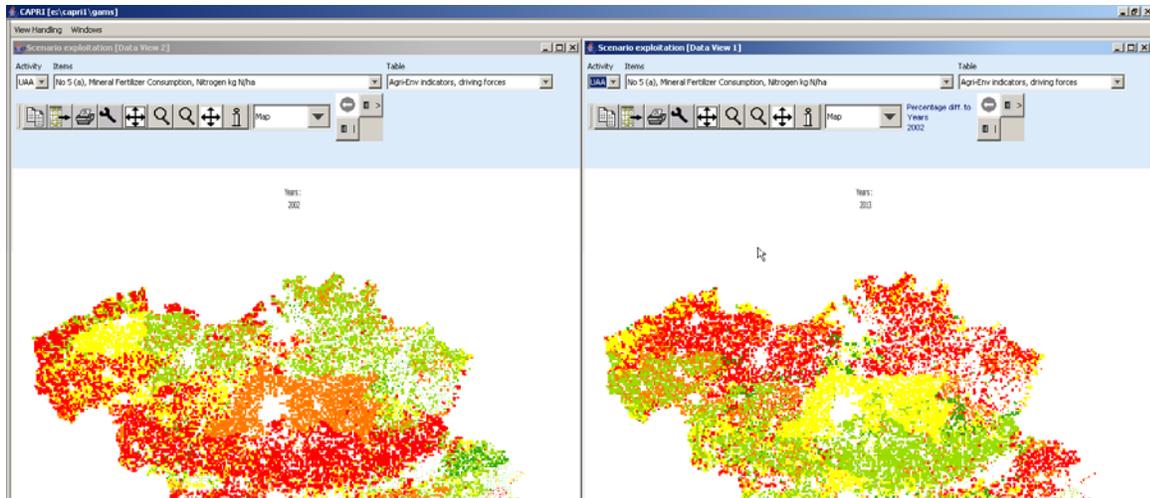
Element used for comparisons: 2002

OK

Now we should get a result as below.

2002		2013	
H2865	154.99	H2865	12.11%
H2866	195.89	H2866	87.83%
H2867	122.77	H2867	82.29%
H2868	119.37	H2868	73.48%
H2869	95.88	H2869	23.66%
H2870	98.81	H2870	25.39%
H2871	86.37	H2871	94.25%
H2872	77.68	H2872	65.59%
H2873	155.88	H2873	16.99%
H2874	75.71	H2874	128.99%

Now, for both views, the output should be switched to maps, and there we are ...



## **The software behind the mapping viewer and the CAPRI exploitation tools**

Reading the following chapter is not necessary to work with the GUI, but more directed towards to technically interested reader.

The original software implementation of CAPRI was based on software available at ILR at that time and comprised a DBMS realized in FORTRAN with C/C++ code for the GUI. The very first maps in CAPRI (in 1998) were produced with an MS-EXCEL mapping plug-in which was at that time a cost-free add-on. However, moving the data to EXCEL and then loading them in the viewer was not a real option for the daily debugging work on data base and model. Therefore, shortly before the first CAPRI project ended in 1999, a JAVA applet was programmed by W. Britz which was able to draw simple maps from CSV-Files automatically produced by the CAPMOD GAMS code. That code with slight modification remained active for quite a while, and some of the features are still to be found in the current mapping viewer. Then for a while, the exploitation tools were based on XML/XSLT+SVG and a mapping viewer in SVG was realized. However, the XML solution had the big dis-advantage of requiring a large amount of single ASCII input files, and was not really performant when complex pivoting was used.

Therefore, the next evolution step was a pure Java GUI with direct access to GDX files which is the current state of the art. GDX files are an internal file format used by GAMS which allows a rather efficient I/O for large sparse tables. An API library allows to access GDX files from other applications. That design has the obvious advantage to be firstly based onto the portable JAVA language. Secondly, as no external DBMS is used, it is possible to use CAPRI by solely executing GAMS programs. It is hence possible to use CAPRI on any systems where GAMS can be run, without the need to install additional software.

The GUI consists of three rather independents components. Firstly, a GUI to control the different work steps of CAPRI. The code deals mostly with defining GUI controls (button, scroll-down lists etc.) to manipulate properties of CAPRI tasks, and then to start them as GAMS processes. That part has been thoroughly refactored with the revision 2008. A second important part is the CAPRI exploitation tool, which are basically generic enough to be used for other modeling system as well. The current refactoring left most of the code untouched compared to the code based developed since 2006, with the exemption of the graphics which is now based on the JFreeChart library. However, as discussed below, in 2007, the mapping viewer was refactored in larger part to host the 1x1 kim grid solution developed in the CAPRI-Dynaspat project. The exploitation tool is a rather unique solution to exploit result sets from economic models based on the definitions of views which are defined in XML tables. It combines features from

DBMS reporting, data mining, spreadsheet functionalities and GIS into one package. And thirdly, there are some specialized pieces as the HTML based GAMS documentation generator which are linked into the GUI.

### ***CAPRI tasks as business model***

A core concept in the new layout is a business object called *AgpTask*. Technically defined as an interface, such an object represents a work task in the overall CAPRI system such a run of CAPREG to build the regional data base. The interface requires getters and setters for properties such as *baseYear*, *simYear* or *MemberStates*. The setters can be accessed either by a GUI interface or by the batch execution facility, formally by class implementing the interface *AgpTaskHandler*.

Most tasks are GAMS executable tasks according to their *isGams* property. These tasks also provide access to the name of the related GAMS program via *getGamsProgramName*. Each of these tasks has also a method called *generateIncludeFile* which generates the specific GAMS include file for that task.

The objects also know about the main GDX file they are generating via *getGdxResultFiles*. Related to that, they allow setting the logical names of the data dimension in the result data set via *setDimNames* and *setXMLTablesDims*.

Once the properties of a task had been defined, their logical consistency can be checked by invoking the method *checkSettings*. *Check settings* returns a string with a description of the first error encountered.

That layout eases dramatically the update process of CAPRI. Definition of new tasks or changes to existing ones will generally not require changes in the GUI, but simply either implementing a new object with the required methods or updating an existing one.

### ***Execution of tasks via a GamsStarter and GamsThread***

Execution of tasks with the property *isGams* is handled by a *GamsStarter* object. An instance of *GamsStarter* let the task write out the necessary include file(s) to generate a specific instance of the specific task (a simulation run for a specific scenario, simulation year, with the market model switched on or off ...). A *GamsStarter* also knows about the working directory or other specific GAMS settings as the scratch directory. It may generate a pipe for the GAMS output to the console to show it in a GUI.

An *AgpTask* can be executed by a *GamsStarter* who will create then a *GamsThread*. A *GamsThread* extends the *SwingWorker* interface of Java so that it may communicate with the normal event queue of JVM. A *GamsThread* can be gracefully terminated by sending a a SIGINT signal to the GAMS process. That will let the GAMS execution stop at specific point determined by the GAMS engine itself and start the finalisation handling of GAMS as removal of intermediate files and directories.

## ***Refactoring the mapping part***

When the 1x1 km grid layer was added to CAPRI during the CAPRI-Dynaspat project it became obvious that the existing JAVA code to produce maps needed some revision, especially regarding the way the geometry was stored. In that context the question of using an existing GIS independently from CAPRI or the use of existing GIS classes plugged-into the CAPRI GUI was raised again; and some tests with open-source products were undertaken. A stand-alone GIS as the sole option was certainly the less appealing solution. Firstly, it would have required producing rather large intermediate files, and would have left the user with the time consuming and often error prone task of exporting and importing the data. Secondly, the user would need to switch between two different programs and GUI standards. And thirdly, all the usual problems with installing and maintaining additional software on a work station would occur. However, as indicated later, the GUI naturally allows passing data over to external application and does hence not prevent the user from using a full-fledged GIS solution.

The main points taken into account during the search of a map viewing solution for CAPRI were: (1) possibility to import efficiently data from the CAPRI GUI, (2) user-friendliness, (3) performance and (4), in the case of plug-in libraries, expected realization and maintenance resource need, and naturally (5) license costs. It turned quickly that an ideal product was not available. Some of the products were not able to allow for the necessary link between newly imported tables with region codes and an existing geo-referenced geometry. Others had very complex user interfaces or produced run-time errors, took ages to draw the HSMU maps or very quite expensive. From the different option tested, the gvSIG (<http://www.gvsig.com/index.php?idioma=en>) freeware GIS seemed to be the only allowing the user to import data from a CSV – which must however be semi-colon delimited – and join one of the columns to a shapefile. At least the version installed at that time was however running not very stable.

In the end, it was decided to build on the existing code base and let Wolfgang Britz write the additional code “on demand”. The main advantage of that approach is the fact that the mapping view is transparently integrated in the CAPRI GUI, it is sufficient to switch from “Table” to “Map” in a drop-down list to produce a colored map, and that user demands regarding additional functionality may be and had been added, taking into account the specific needs of the CAPRI network.

Compared to ArcGIS, where the EU27 HSMU geometry plus codes and centroids requires about 340 Mbytes, the CAPRI version requires about 27Mbytes solely. Reading in the CAPRI GUI is somewhat slower compared to ArcGIS due to unzip on demand. The actual drawing operation takes about the same time as in ArcGIS (about 11 second for the full data set), classification in Java is typically faster.

## ***Views as the basic concept***

The concept of the CAPRI exploitation tools is centred on the idea of a view. Content wise, that may be understood a certain angle to look at European agriculture as example its integration into international markets, its impact on the environmental, farm management, costs of the Common Agricultural Policy. Each view thus extracts a certain collection of numerical values, labels them so that they carry information to the user (long texts, units), chooses a matching presentation – as a table, as a map or a graphic, and arranges them in a suitable way on screen. The views can be linked to each others, allowing a WEB like navigation through the data cube. The user may have open several views in parallel, and he may change the views according to its needs, e.g. switch from a map to a tabular presentation, or change the pivot of the table, sort the rows etc.

Internally, each view is stored in a XML schema. A view could be also understood as a combination of a pre-defined selection query, along with reporting information. The XML schema allows to attach long texts, units and tooltips to the items of a table, and thus to show meta-data information to the user. The XML schema does hence replaces look up tables in a DBMS. It may equally store information regarding the pivoting, the view type (table, map, different graphic types), and for maps, classification, color ramp and number of classes. The views can be grouped into logical entities, and are shown as a popup menu to the user.

Tabular views may feature column and row groups. Empty columns and rows can be hidden, tables can be sorted by column, with multiple sort columns supported. Numerical filter can be applied to columns.

## ***Data model***

The underlying data model is very simple and straightforward. All data are kept in one large multi-dimensional data cube, and all values must either be float values or strings. Currently, only read-only is supported. Each data dimension is linked to a vector a string keys. Those keys are the base for the filter definitions. Currently, data cubes up to six dimensions are used (regions – activities – items – trading partners – years – policy scenarios). The data storage model is equally optimised to the specific needs. As only float values or strings are supported, all data can be stored as one primitive array of either floats or strings. To allow fast and efficient indexing, a linear index is constructed from the multi-dimensional data cube, and the non-zero data and their indices are stored in a hash table. That renders data retrieval very fast. All data are at initialisation time loaded in memory, for moderately long linear indices, about 10 Bytes are required to store a non-zero float and its index as an int. If the maximal linear index is very large, the index is store as a long and the storage needs goes up to about 16 Bytes. For moderately sized data cubes, 20 Million numbers can hence be hosted in about 200 Mbytes.

The data are read from a generic file format generated by GAMS (General Algebraic Modelling System, a commonly used software package in economic modelling) called GDX, the software package on which CAPRI is based. Access to GDX is handled via an API provided by GAMS.

### ***Client based solution***

Technically, the exploitation tool is completely client based. That reflects the specific user profile of the CAPRI modelling system where the exploitation tool is integrated with an economic model and tools building up its data base. The main aim of the tool is to support forward looking policy analysis, and user will create their own scenarios and in some cases even own variants of the export data, processes requiring considerable processing and storage resources. A client-server solution where the production process and data storage would need to be hosted on a web server is therefore not a preferred solution, also as users will often develop variants of the modelling system by code modification in GAMS, and contribute to its development. The structure of the data driver would however very easily support linkage to network or WEB based data bases. It should however be noted that the data base and GAMS code are managed via a Software versioning system, which is a kind of client-server environment.

### ***The geometry model***

The mapping viewer of CAPRI is based on very simple and straightforward concepts. First of all, it supports solely polygon geometries not comprising holes. The storage model is optimised to host rectangles, and is especially efficient if the polygons vertexes are all points in a raster. The topology is not read from a shapefile, but stored in a generic rather simple format. The vertices are stored in x,y coordinates already projected in a rectangular coordinate system, and the viewer does not deal with geographic coordinate system, but simply scale the rectangular coordinates in the viewport. The viewer in its current version solely supports one layer of quantities. Those restrictions naturally allow reducing memory needs, and, thanks to the rather simple data structures, also allow rather performing drawing operations. It should be noted that the JIT compiler of JAVA is indeed rather fast.

The biggest topology currently handled simultaneously covers an intersection of Corinne Land Cover, slope classes and Soil Morphological Units and comprises for EU27 around 2.7 Million polygons. As the majority of the polygons are rectangles, not more then 6-7 Million points needed to be stored. The topology handler and the drawing routines separate rectangles, for which only the two outer points are stored, from polygons, for which the vertices and centroids are stored.

The viewer is written in Java. There are two variants. One is a stand-alone version of the viewer realised as an applet. It reads from an internal portable binary data format, and java classes, data and geometry can

be packed into one jar file, e.g. to ship it to a client. The second version is transparently integrated in the GUI of the CAPRI modelling system.

Swing is used for the GUI. In order to profit to the most of the simple implementation, the viewer has been written completely anew, and is not based on existing GIS libraries. Even certain standard JAVA classes as e.g. for hash tables, have been replaced by own implementations, to reduce implementation overhead. Some care was given to support flexibility in classification, given that only quantities are supported, so that the tool covers natural breaks, quantiles, equal spread, mean standard and nested means. Area weighting is supported as well.

In order to export data to other applications, the tools supports first of all tab delimited clipboard export, allowing import e.g. into EXCEL. Maps can be exported as JPEGs over the clipboard. Alternatively, the user may export to external file, in CSV format, DBF, to MS Access or to GAMS. DBF export will generate a second file comprising meta data.

The exploitation tools of CAPRI build on a rather simple structure. Each CAPRI work step stores its results a GAMS parameter representing a multi-dimensional sparse cube which is stored as a GDX file. The exploitation loads the non-zeros from one or several GDX files into memory. However, given the length of the different dimensions and the use of short codes, the user would be typically lost on its own in the large tables. The XML definition file is the equivalent of a collection of “SQL queries” as it defines views which combine filters in the dimensions of the cube with information how to show the result (pivot; table, graph or map).

# The structure of the XML definition files for the views

## ***General comments***

It is not intended to let the user edit that file, but in order to have a complete documentation, some information about the structure is included in here. The XML parser used by the GUI's java code is not general XML parser as test revealed that the java base general XML parsers were rather slow. For the XML file used for the definition the views (the standard name is "tables.xml"), using a simple parser has some consequences: only one tag is allowed per line, and tags are not allowed to span several lines. Also, error handling is so far only rudimentary as users are not supposed to edit that file.

The table viewer supports currently up to 6 dimensions, which are named as:

1. Region
2. Activity
3. Product
4. Scenario
5. Year
6. Dim5

in the XML-file. These "logical dimensions" can be mapped to any dimension of the original data cube read in by the java code. Pivoting can then be used to map these "logical" dimensions to viewport dimensions seen by the user such as the columns and rows of a table.

## ***Necessary tags for tables***

A table definition is found between the `<table> ... </table>` tags. It must at least define:

- The table theme, such as `<theme>Welfare</theme>`. The themes are shown as a drop-down menu in the exploitation tools.
- The table name, such as `<name>Welfare comparison between Member States</name>`. The names must be unique.
- The items of the tables.

The order of the themes and table names will define their order in the drop-down menu.

## Defining the items of the table

The underlying idea of having for one of the definitions a “hand defined” list of items stems from the observation that most tables have only a very limited number of columns, and that these are normally formatted with care regarding their text comprised. Each table requires therefore a definition of items, but the items must not be necessarily mapped in the column viewport.

`<item>`

`<itemName>Money metric</itemName>`

`<key>CSSP</key>`

`<unit>Mio Euro</unit>`

`<longtext>Consumer welfare measurement: expenditures necessary to reach utility in current simulation under prices of reference scenario</longtext>`

`<link>Money metric</link>`

`</item>`

An item definition is enclosed in the `<item>...</item>` tags. It must at least comprise a `<key>` and an `<itemName>` tag. The case sensitive key must match the symbol identifier as found in the GDX file, whereas the `itemName` can be freely chosen.

Facultative tags are:

`<unit>`: a physical unit shown in table

`<longtext>`: a text shown when the mouse hovers of the column

`<link>`: a link to another table for the table cells under the column.

`<colormode>`: the color mode used when a map is drawn for the item. The following modes are supported:

- GYR Green Yellow Red
- RYG Red Yellow Green
- GR Green Red
- RG Red Green
- BG Blue Green
- GB Green Blue

- WB White Black
- BW Black White
- LD Light Blue Dark Blue
- DL Dark Blue Light Blue

**Items can only stem from the product or activity dimension.** In order to define from which dimension they are taken, the user can set either:

`<isActivity>NO</isActivity>`

Which means that the tables loops over the products, and the items refer to the activity dimension. A typically example is a table with market balance elements: the items such as “FEDM” are found in the columns of the CAPRI tables where also the activities are stored. Consequently, the table will loop over the products, and not over the activities. Alternatively:

`<isActivity>YES</isActivity>`

allows only items from the product dimension, and let the table loop over the activities. A typically example are the table shows activity levels, yield or economic indicators for the production activities.

## **Additional tags**

`<defpivot>`

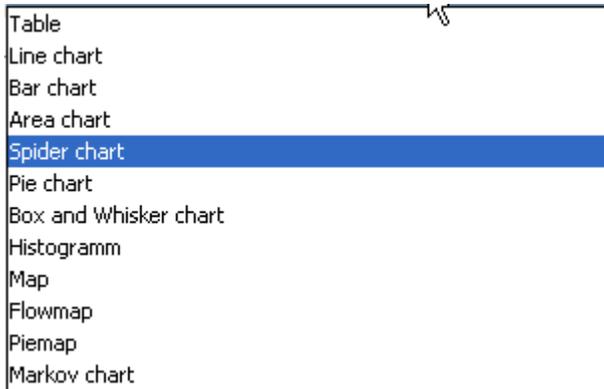
Defines the default pivot used for the table. The pivot string consists of characters. The first character position is for the table row blocks, the second for the table rows, the third for the column blocks and the last for the columns. The logical dimensions are labelled with the following characters:

- R regions
- P products
- I Items
- S Scenario
- D Dim5
- A Activity

The definition `<defpivot>0R0S</defpivot>` thus means: regions are in the rows, scenarios in the columns. The definition `<defpivot>PISR</defpivot>` puts the products in the row blocks, the items in the rows, the scenarios in the column blocks and the regions in the columns.

## <defview>

Defines the default view used for the tables, the list of default views is equal to what the user can select in the drop-down box:



## <COO>

This tag defines the geometry to use for maps. Currently, the following geometry files are available:

- NUTSII.zip    NUTS 2 geometry for countries covered by the supply module
- MS.zip        NUTS 0 geometry for the countries covered by the supply module
- RMS.zip        Global geometry for the regions with behavioural functions in the market model
- RM.zip        Global geometry for the trade blocks in the market model
- HSMU.zip      1x1 km pixel clusters for EU 27 without Malta and Cyprus

There are also 1x1 km pixel clusters for individual Member States, but these are internally passed to the viewer when only one country is shown.

## Alternative texts for the dimensions

Normally, the names for the dimensions are passed in the view by Java. However, their name can be changed by:

<regionText>...</regionText>

<activityText>...<activityText>

<regionText>...<regionText>

<productText>...<productText>

<scenText>...<scenText>

<dim5Text>...<dim5Text>

<yearText>...<yearText>

That text is shown:

- As description above the outer drop-down selection boxes:



- In the pivot dialogue:
- And in graphics / map titles and the like.

## ***Filters for the elements in the different dimensions***

Without filters, all elements found on a logical dimension will be shown to the user in any table. The exemptions are the items either defined for the product or the activity dimension, see above. In order to restrict the selection in the other logical dimensions, selection list can be defined in the table definition. Take as an example the following XML tag:

<regionSel>MS</regionSel>

It means that the table will only show elements with the tag <region> (see below) which comprise MS in their <sel> field. The example would refer to the Member States. There is a specific selection list:

<regionSel>FromDataCube</regionSel>

Which will neglect the elements under <region> as defined in the file, but rather takes any one found one in the data cube. The option was introduced to avoid the necessity to define all 180.000 HSMU codes in the file.

## ***Attaching long texts and filters to elements***

Items for activities, products, regions and dim5 are typically defined in the file, see the following example:

<region>

<key>SK020038</key>

<itemName>SK020 - FT41 / GT100 - Specialist dairying (FT 41)</itemName>

```
<sel>[all, RS, SK, FA, SKFA, FT41, GT100, FT41GT100]</sel>
```

```
</region>
```

The definitions for one item are enclosed in the tag (`<region>...</region>`, `<activity>...</activity>`, `<product>...</product>`, `<dim5>...</dim5>`).

The order of the items in the tables is defined by these lists.

Each item has a `key`, which corresponds to the symbol identifier found in the GDX file. The keys are case sensitive. The `itemName` is a long text which is typically shown to the user. The elements found between the `<sel> ...</sel>` tags can be used as filters in table definitions, or interactively by the user.

A specific tag is `<aggreg>yes</aggreg>`: when found for an item in the rows, it will be shown twice in the table: once in the top part, and then again.

# Index

- CAPRI installation 11
- Changing fonts 43
- Column and row selection 39
- Comparison output 43
- drag 80
- drawing several maps 112
- file export 79
- Flow maps 64
- full extent 80
- Gams settings 11
- Graphics
  - Bar charts 56
  - Box and whisker charts 61
  - Clipboard export 56
  - Export to file 55
  - Histograms 62
  - Line and point charts 57
  - Markov charts 63
  - Pie charts 59
  - Spiderweb plots 60
- Hiding empty columns or rows 43
- Histogram 44
- info pointer 80
- legend
  - continuous linear scaling 77
  - continuous logarithmic scaling bar 78
- Login in 10
- map with percentage changes 109
- Maps
  - Classification
    - Area weighted classification 70
    - Classification method 70
    - Color table 73
    - Equal interval 71
    - Excluding zeros from classification 70
    - Manual classification 72
    - Mean standard dev 72
    - Natural breaks 71
    - Nested mean 72
    - Quantile 71
  - clipboard export 79
  - Frequency diagram in map 73
  - Getting data for specific polygons 80
  - Highlighting specific regions in the map 81
  - Histogram window 68
  - Info pointer and window 80
  - Regional labels in map 85
  - Shrinking polygons according to UAA share 69
  - Updating the map 85

Number format and rounding 43

numeric filter 83

Online help 13

Pie chart maps 66

Pivoting 41

Tables

Drill down 46

Filtering 47

Outlier detection 50

Sorting 47

Statistics 48

title of the map 79

View

scenario result 100

View data

Base year 99

baseline 100

View Selection 38

View type selection 41

zoom in 79

zoom out 80