

VEDA[®]

(Version 4.3.8)

KanORS Inc/HALOA Inc
Montréal, Canada

October 2003

VEDA (VERsatile Data Analyst) is a flexible, user-friendly tool for the construction of report tables and graphs from GAMS-based models.

[®] Veda [Sanskrit,=knowledge, cognate with English wit, from a root meaning know], oldest scriptures of Hinduism and the most ancient religious texts in an Indo-European language. The authority of the Veda as stating the essential truths of Hinduism is still accepted to some extent by all Hindus. The Veda is the literature of the Aryans who invaded NW India c.1500 B.C. and pertains to the fire sacrifice that constituted their religion. The Vedic hymns were probably first compiled after a period of about 500 years during which the invaders assimilated various native religious ideas. The end of the Vedic period is about 500 B.C. Tradition ascribes the authorship of the hymns to inspired seer-poets (rishis). [The Columbia Encyclopedia, 6th ed. New York: Columbia University Press, 2000.]

Table of Contents

INTRODUCTION	5
1 INSTALLATION	7
2 GETTING STARTED	8
2.1 Overview of VEDA features and main menu	8
2.2 Importing scenarios into VEDA	10
2.3 Setting defaults	11
2.4 Creating an Analysis Table	13
2.4.1 Description of the Main VEDA Screen (Figure 1)	13
2.4.2 Defining the contents of a new table	16
2.4.3 Specifying the table layout	20
2.4.4 Aggregation rules	21
2.4.5 Miscellaneous functions	21
2.5 Viewing a Table: the Cube view	22
2.5.1 Altering table layout in the <i>View</i> mode	24
2.5.2 Altering table contents: Removing elements from a dimension	26
2.5.3 Exporting result tables	27
2.5.4 Graphs	27
2.5.5 Print	28
3 ADVANCED FEATURES	29
3.1 The batch mode	29
3.2 User sets	29
3.2.1 Creating a new set	30
3.2.2 Set Updating (Refresh)	33
3.2.3 Exporting and Importing sets	33
3.2.4 Viewing and editing user defined sets	33
3.3 Aggregation Rules	34
3.3.1 Expression Builder and Collapse operations	34
3.3.2 Enumerate	36
3.4 Viewing the Reference Energy System (for the MARKAL family of models)	36
4 Reference Section (Main Screen Menu)	40
4.1 File Menu	40
4.1.1 New database	40
4.1.2 Open database	41
4.1.3 Compact database [F12]	41
4.1.4 Backup database	41
4.1.5 Restore database	41

4.2	View menu	41
4.2.1	Toolbar	41
4.2.2	Statusbar:	42
4.2.3	Open folder(s): Activates VEDA Table folders.	42
4.2.4	Show/Hide Tabs	42
4.2.5	Reference Energy System (Only for the MARKAL Family of models)	42
4.2.6	Execution Mode	42
4.2.7	Search [F3]	42
4.2.8	Refresh [F5]	42
4.3	Table menu	43
4.3.1	New Table [Ctrl+N]	43
4.3.2	Save Table [Ctrl+S]	43
4.3.3	View Table [Ctrl+L]	43
4.3.4	Delete Table [Ctrl+D]	43
4.3.5	Rename Table [Ctrl+F2]	43
4.3.6	Import Table(s)	43
4.3.7	Export Table(s)	43
4.3.8	Table Master [Ctrl+M]	43
4.3.9	Delete Cube File(s)	43
4.3.10	Sort Tables	43
4.4	Sets menu	43
4.4.1	New Set(s) [F6]	43
4.4.2	Edit Set	45
4.4.3	Delete Set(s)	45
4.4.4	Examine (View) Set	46
4.4.5	Refresh Sets [Ctrl+F5]	46
4.4.6	Import Set(s)	46
4.4.7	Export Set(s)	47
4.4.8	Copy Across Parents	47
4.5	Results menu	48
4.5.1	Import/Archive... [F7]	48
4.5.2	Import Topology Data [for MARKAL models only]	50
4.5.3	Select Database	50
4.6	Tools menu	50
4.6.1	Reset Cube Settings	50
4.6.2	Set Default Layout	51
4.6.3	Delete All Cube Files [Shift+Del]	51
4.6.4	Options [Ctrl+O]	51
4.6.5	Units	56
4.6.6	Reset database [Ctrl+F12]	56
4.6.7	License Operations	56
4.7	Help menu	56
4.7.1	About Veda	56

4.7.2	Contents [F1] _____	56
4.7.3	Search [Ctrl+F1] _____	56
4.7.4	KanORS.com [Shift+Ctrl+F1] _____	56
<i>Appendix A: GDX2VEDA Data Definition File</i>		<i>57</i>
<i>Appendix B: VEDA MARKAL/SAGE/TIMES Attributes</i>		<i>60</i>
<i>Appendix C. VEDA-BE versions</i>		<i>61</i>

Introduction

The **VE**rsatile **D**ata **A**nalyst is a software dedicated to the analysis of data and results obtained from a broad variety of mathematical models or data bases. VEDA is a powerful and user friendly tool for the construction of analyst tables and graphs to help in the analysis of results from complex mathematical models.

VEDA is suitable for any GAMS-based model built on the concept of Sets for identifying and grouping model components and parameters for the management of numeric values, including those provided by the user to the model, derived in GAMS code as part of the model preparation and/or reporting procedures, as well as those obtained from the model solution (e.g. the primal and dual solution values obtained from optimization). Any GAMS-based model can be linked to VEDA by means of the GDX2VEDA utility and the associated <application>.VDD VEDA definition file. Appendix A discusses the GDX2VEDA Data Definition File in some detail.

In this manual, examples are taken from the World SAGE¹ model, which is one variant of the generic MARKAL family of models². The manual is also fully compatible with the TIMES model as well, and illustrative of how VEDA can be used with most GAMS model.

Besides its generic capabilities, VEDA has additional features to exploit the MARKAL family of models relying on a general network structure, where nodes represent *processes*, and the arcs linking these nodes together represent *commodities* produced and/or consumed by these processes. In the case of energy models, such a network is called a Reference Energy System (RES), and the commodities are the various energy forms, materials, and emissions (existing or future) that are produced and/or consumed by energy processes, as well as the demands that the system must service. However, as already noted the realm of application of VEDA extends to any GAMS-based model. Any such mathematical model possesses at least some of the following elements:

- **Sets**, which are collections of common components (e.g., all central station power plants, fossil energy carriers, residential demands);
- **Parameters**, which are data inputs (e.g., technology characterizations, resource supply curves) and intermediate calculations (e.g., matrix coefficients) of the model;
- **Variables**, which are the quantities computed by the model (e.g., level on new investments, the activity of processes);
- **Equations**, which are mathematical relations between expressions involving variables and attributes (e.g., commodity balance), and
- **A Reference System** indicating the network structure (the topology as depicted by the input and outputs from the various processes) of the model.

¹ *System for the Analysis of Global Energy* markets (SAGE), Model Documentation, US Energy Information Administration, 2003 -Volume 1, <http://www.eia.doe.gov/bookshelf/docs.html>.

² The MARKAL family of models consists of a number of variants of MARKAL itself, SAGE a special time-stepped version, and the new TIMES model. See www.etsap.org for more information on each of these models.

In order to apply to a particular model, VEDA must be provided with the list of sets corresponding to the indexes into parameters, variables, and equations, the list of relevant parameters, variables and equations (for some models, results concerning equations may be absent) to be managed. VEDA then organizes and provides access to the various data forms in a transparent manner, allowing the analyst to focus on exploring and interpreting the meaning of the information. Each of the indexes into this data is presented on a separate “Tab” on the main VEDA table specification form, and corresponds to the various row qualifiers, as well as the nature of the data column(s), of the resulting analysis tables constructed by VEDA. Each kind of numeric information managed is named by the user and presented on one of the Tabs of the VEDA table specification form (for the MARKAL family of models this corresponds to the Attribute tab).

This user’s manual assumes that the model results have been processed by a model-specific GAMS program (discussed in Appendix A) that produces data files in a format suitable for input into VEDA. Such data files contain all relevant information on the sets, variables, attributes, and equations, and on the Network (Reference System), if appropriate, on which the model is based. Additional information on linking GAMS models to VEDA is available from Gary Goldstein at (ggoldstein@gams.com).

The user creates analysis tables by choosing their contents, and refining their layout. The software may function in interactive or batch mode. This user’s manual is divided into four sections and three appendices:

- Installation of the VEDA software;
- Getting started;
- Advanced features;
- Reference Manual;
- Appendix A: GAMSdata definition file GDX2VEDA
- Appendix B: List of MARKAL attributes
- Appendix C: VEDA versions

It is recommended to get fully familiar with the Getting started features before trying the advanced features of VEDA. The Reference Manual contains details on all the VEDA menu options.

Note also that a sister software called VEDA-FE (front-end) allows the browsing and manipulation of input data for the MARKAL/SAGE/TIMES models. But unlike VEDA-BE, VEDA-FE is a system dedicated to each particular model and may not be readily adapted to support other models, GAMS-based or not.

1 INSTALLATION

Access the Download page at the web address: <http://www.kanors.com/ETSAPdnld.htm>

The following page appears. First download the msi file. Then follow the instructions. If necessary, you may update the VEDA.EXE version at any time by downloading the latest version (here version 4.3.8 is displayed).

Download VEDA4 for MARKAL/TIMES

[Veda-4.3.2](#)

GAMS Code

Installation (msi file - 8.56 Mb)

MARKAL code version 5.2a (zip file - 428 Kb)

EXE Updates

[Veda-4.3.8](#)

from version 4.3.1 and above

Installing VEDA4

- Execute the file VEDA4.msi. *You might encounter Error 1931 during the installation process; click OK.*

If your computer is connected to the internet:

- Launch VEDA4; click the button "Get key online..." on the Registration form that pops up.
- Enter the email address where you received the information about this page, and click "Generate".
- You will receive a license file by email; save it in the VEDA folder.

If your computer is NOT connected to the internet:

- Launch VEDA4, and click "Exit" on the registration form.
- Goto <http://www.kanors.com/licVeda4.asp>
- Enter the email address where you received the information about this page, and the first 16 char string from the file VEDALic.txt in the VEDA folder; Click "Generate".
- You will receive a license file by email; save it in the VEDA folder.

Launch the application again, and enter your personal information. Click "Register me now." *This completes the VEDA installation.*

2 GETTING STARTED

This section introduces the reader to basic operations required to create an analysis table, starting from scratch. The following steps are involved: Import scenarios, Define new table(s), View, and then Save/Export/Print the table(s) thus created.

2.1 Overview of VEDA features and main menu³

VEDA requires as input four text files containing full information on each model run that the user wishes to examine and process. These files are produced by the GAMS GDX2VEDA utility. The generic names and role of these files are as follows:

- <scenarioname>.VD, containing the application VEDA header directives (controlling the appearance of the main VEDA table specification form) and actual mode data;
- <scenarioname>.VDE, containing the list of individual set member elements for each index managed by VEDA with their descriptions;
- <scenarioname>.VDS, where the set grouping information is provided for the various sub-sets associated with each index managed by VEDA, and
- <scenarioname.VDT>, containing the topology (RES) information, for MARKAL-based models only.

These files must be present in a user chosen folder in order for VEDA to be able to **import** the corresponding scenario (see section 2.2). Once the scenario results are imported, VEDA offers the main menu briefly outlined below (section 4 contains a complete menu description).

File: allows several operations on the databases being examined: ***New, Open, Compact, Backup, Restore, and Quit***.

View: besides the usual view choices such as showing or hiding parts of the screen, this menu item has three important functions: a) to allow the user to select subsets of tables in specific ***folders***, b) to switch to the graphic Reference Energy System (***RES***) ***view (for MARKAL-based models)***, and c) to choose the ***execution mode*** (interactive or batch). See section 3 for these advanced features.

Tables: a table is the basic unit consisting of rows and columns created by VEDA from the data imported. This menu allows the user to define a ***new*** table, to ***save*** or ***erase*** a table, erase ***Cube files*** (the final form in which a table is organized), and set the default layout of a table. See sections 2.3 and 2.4 for descriptions of these features.

Sets: sets of elements (e.g., commodities and processes for MARKAL-based models), grouped according to some logical condition defined by the model itself or by the user. Sets are often very useful to define the contents and organize the layout of a table. Some sets are permanently defined by the model (e.g., for the MARKAL family of models,

³ A complete description of Menu options is provided in section 4 of this manual.

TCH = all technologies, ENC = energy carriers, MAT = materials, DM = demands, etc.), and may be freely used in VEDA. In addition, this menu allows the user to create **new** sets, to **edit**, **erase**, or **refresh** existing sets, to **import** sets defined by another user, and to **export** sets to another location. See section 3 for all operations on sets.

Results: as already mentioned, results must be imported into VEDA after the completion of a model run, before tables may be created. This menu allows **importing** (and setting of the import location), **retrieving**, **archiving** and **deleting** results, as well as defining the directory where result files are stored. See section 2.2.

Tools: this menu offers a number of **options** such as: **formatting** of table entries (e.g. row/column names, cell formats, table titles), choosing the **default directories** where input and exported files are stored, defining a default table layout, etc. These options will be discussed in sections 2 and 3 when needed. The **Units** selection allows for the definition of new units and to assign default units to the data components (called attributes in this manual). After units have been defined and later assigned to tables when they are constructed, conversion factors may be stored and applied so that the resulting tables may be presented in any desired alternate units instead of the default units.

Help: a complete help facility offers general VEDA information, as well as a function oriented and a search engine for various VEDA features.

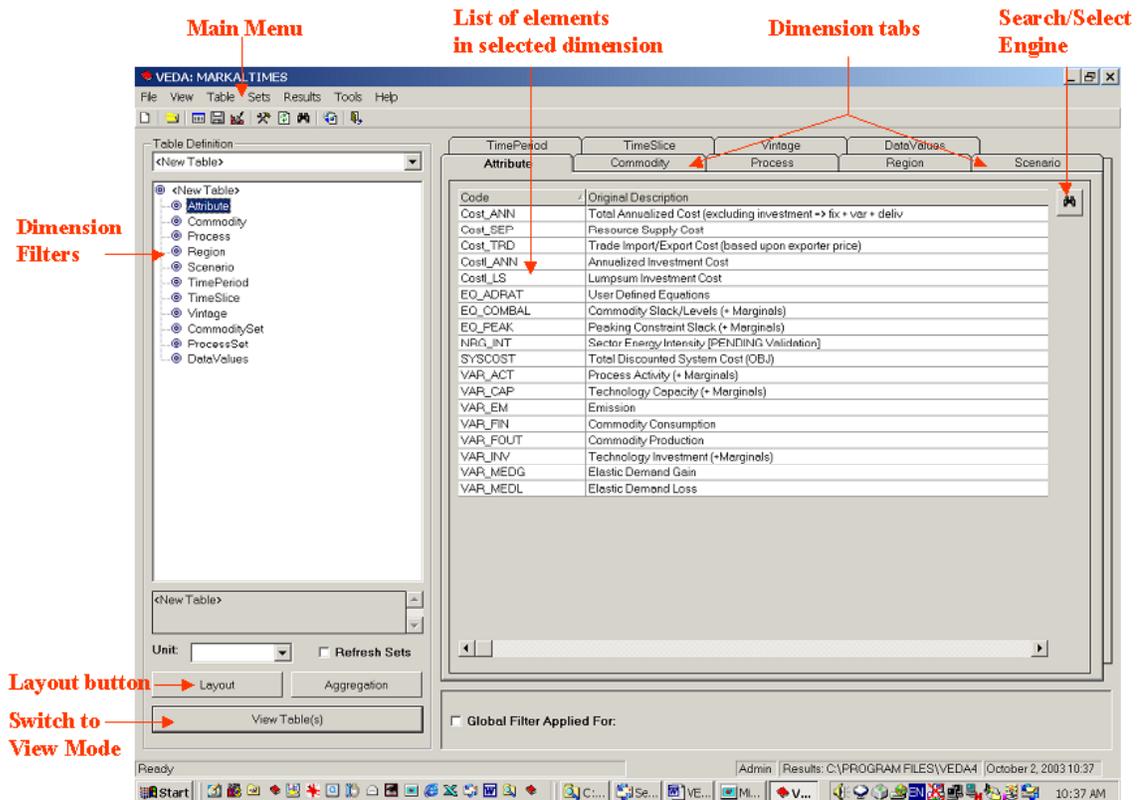


Figure 1. VEDA main screen

2.2 Importing scenarios into VEDA

In order to analyze the results of a model run, you will need to import the scenario files into VEDA. First, run the model and put the GAMS result text files (<scenario>.vd; <scenario>.vde; <scenario>.vds, and <scenario>.vdt, produced by the GDX2VEDA utility) in a directory of your choice, then declare this default location to VEDA, using the following steps:

To import file(s) containing scenario data and results

First specify the location of the data files as follows:

- From the Tools menu, choose *Options* (or press Ctrl+T)
- In the *Options* dialog box, select the *General* tab
- Press the *Manager Window* button or the  button to define one or more paths containing the input files.
- Use the scrolling list to select the desired path (if more than one path has been defined)
- Click the OK button.

Then import the desired scenario results as follows:

- From the *Results* menu, select *Import/Archive ...* (or press F7)
- Select the *Import* tab.
- Select the check box(es) for the desired scenario(s).

Click the OK button. The import activity may take some time to create the required database(s) depending upon the size of the model and the number of scenarios being imported at one time. Note that if some scenarios had already been imported at an earlier time, and some new ones are now required, it is faster and more efficient to *Delete All Scenarios* from the *Delete* tab and re-import the desired ones.

Figure 2 shows the *General Options* window with the first folder to be searched for the results files displayed in the *Input Files Paths* listbox window.

⁴ This way of specifying the input file location(s) may also be performed directly on the Results Import tab discussed below.

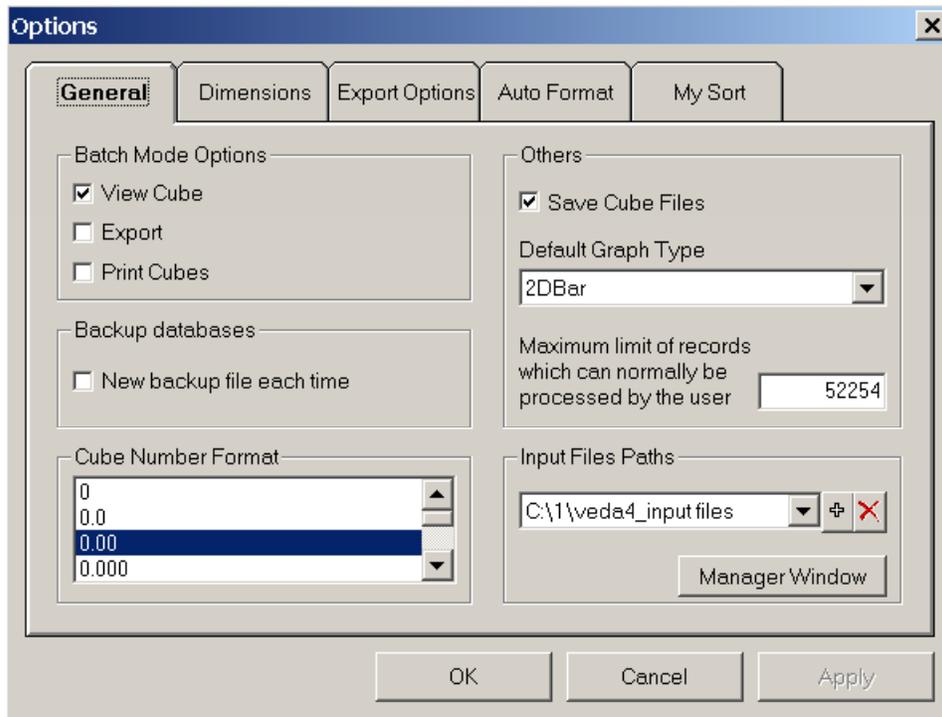


Figure 2. General Options Window

2.3 Setting defaults

Prior to defining and using tables, it is recommended to set a few default preferences to suit your installation and tastes⁵. Setting default options starts with selecting *Tools* in the main menu. Here are a few useful defaults to set. Note that all these defaults may be superseded at later stages of table definition:

In the *Tools/Options/General* tab, choose the number format and whether or not you wish to save cube files (see section 2.5 on VIEW). Saving cube files is recommended since it speeds up the presentation of repeatedly selected analysis tables (when no information is changed in the table since the last request).

Tools/Options/Export Options. In the *Export Path* window, select the folder where you want to save the exported files.. You may also create a new folder into which exported files will be saved by means of the standard *new folder* button (section 2.5 describes the export function). Cube file may be exported in various formats (Excel, Word, HTML, text).

Tools/Options/Dimensions: you may choose whether elements in each dimension are shown according to their (short) code names or by their (long) descriptions. You may also choose whether or not subtotals are shown for each particular dimension.

⁵ Note that in all cases VEDA does provide defaults of all options.

Tools/Options/Autoformat: allows the user to define the default format for all tables (fonts, colors, borders) by right-clicking after having selected the appropriate table portions.

Tools/Options/MySort: allows the user to define their own sorting order for the various Dimension Tabs. This feature is useful when certain dimensions must be listed in a particular order in tables. If defined, My Sort will become an option (along with the ascending and descending sorting orders) while in View mode. See section 2.5.

Tools/Options/Set Default Layout: A default layout to be used when tables are constructed for the first time may be set, and later modified for each specific table if necessary. Figure 3 shows a default layout where there are three levels of nested rows qualifiers (scenario, region, attribute) and one level of columns (time period). The default layout is selected by dragging and dropping the desired dimensions onto rows or columns.

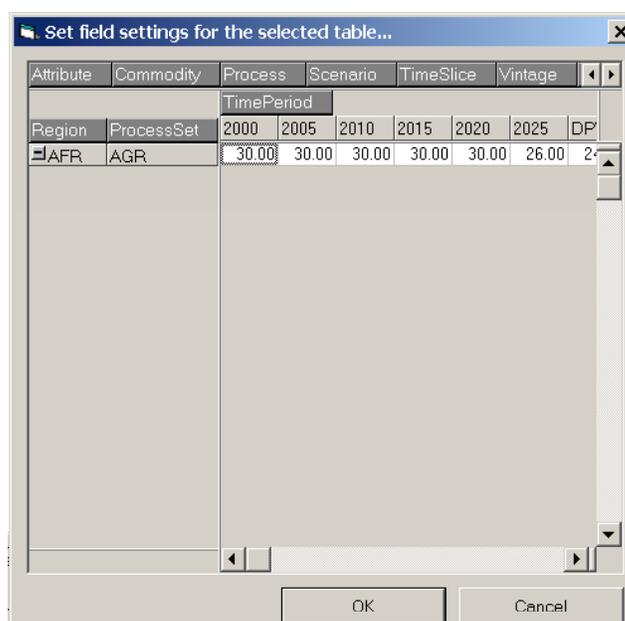


Figure 3. Example of setting the layout of a table using the Layout button

Tools/Units: choose from the main menu the option */Tools/Units/Define-Edit-Delete* to define any units you may need, and conversion factors between them. You may also define default units for each data attribute. Later, when defining a particular table, you will have the opportunity to supersede the default unit if necessary. You will also have the possibility to convert into different units. VEDA comes with a number of predefined units and conversion factors which you may modify or delete at will, as well as define your own new units.

The other default settings relate to organizing your tables and working with multiple databases.

Table folders: in the main menu, choose */Tables/Table Master*. You will be shown a folder structure with the uppermost level called <AllTables>. You may add and delete folders at will, by right clicking and selecting the appropriate action. This feature allows you to create different folders to put your tables in (for instance, you may create one folder per project you are working on). Once the folder structure is created, select from the main menu the option

/View/Open folder, and choose one folder. From now on, and until you change this option, you will have access only to the tables listed in this folder. Note that folder <All Tables> always contains all the tables.

Changing Data bases: by invoking */File/Select System databases*, you may change to another set of databases altogether, provided a folder containing these alternate databases has been created. This feature may be useful if you are using VEDA for two entirely disconnected projects or multiple models.

2.4 Creating an Analysis Table

This subsection describes how to create an analysis table. A table is fully defined in two steps by a) selecting the **contents** of the table --i.e. what results will be presented in the table, and b) selecting a **layout** for the table, i.e. organizing the table's rows and columns. Both steps use the main VEDA screen, shown in Figure 1. We first describe the various windows and buttons of the main screen.

2.4.1 Description of the Main VEDA Screen (Figure 1)

1. **The two table definition windows**: these two interacting windows occupy most of the main screen. They allow the user to select a pre-defined table, or to define a new one, as will be explained in section 2.4.2. The left window is the table-definition window, which contains at its top the list of existing tables (if any have been previously defined) and just below that, the list of dimensions. The user can choose (and then modify) a pre-defined table from the list, or create a new one altogether. Note that the list of tables shown is the one in the folder that has been selected under *\View\Open folder(s)* in the main menu (see 2.3 above). If no folder has been selected, all tables are shown. The right window shows the *Dimensions Tabs*, corresponding to the indexes into the various data attributes, and for the currently selected tab the detailed lists of elements. Note that for the Process and the Commodity dimensions, the *sets* of elements are also shown in a second window above the one showing the elements, as shown in figures 4 and 5. .

Each element in a dimension has a *code* and a *description*. The description comes in two flavors: the original description provided from the model, and the Veda description. You may toggle between the Veda and Original descriptions by right-clicking when cursor in the description column and selecting *Display Type* from the pop-up menu. Initially, these two descriptions are identical, but the user may change the Veda description at will by right-clicking with cursor on one element of the dimension, and selecting *Edit item description*. This feature is particularly useful for scenario descriptions, and also for some set descriptions or region names, in order to adapt the heading of table columns or rows to the terminology of a particular report..

For the MARKAL family of models, the 9 dimensions are:

Attribute: describing the kind of information to be displayed. There are many VEDA attributes handled for the MARKAL/SAGE models, each corresponding either to an elementary model quantity (e.g. capacity of a process), or to a quantity calculated from elementary model quantities (e.g. energy intensity). Figure 1

shows the current list of attributes and their meanings. Appendix B further explains the precise meaning. Future versions of VEDA may have a longer list of attributes.

Commodity and Commodity Set (figure 4): A commodity is either an energy form, a material, an emission, or a demand category. In short, a commodity represents a flow in the RES (whereas a process, as described below, represents a node of the RES). A commodity set is a group of commodities defined by the model or by the user. Commodity sets are convenient ways to avoid having to list many commodities when defining a table. They will be discussed in detail in section 3.

Process and Process Set (figure 5): a *process* is the generic name for a technology or an energy source or an energy/emission sink. A *process set* is a group of processes defined by the model itself or by the user. Process sets are a convenient way to avoid having to list many processes when defining a table. They will be discussed in detail in section 3.

Region: For SAGE, or any multi-region MARKAL/TIMES model, the various regions represented in the model are available. Note that if working with a single region the Region tab may be removed from view (*/View/Show+Hide Tabs*), both as a tab and from all tables. In this case, the tab is only temporarily removed, and may be reinstated later.

Scenario: Each model run, encompassing a complete set of data and results to be managed as a group by VEDA, is given a scenario (or run/case) name. This name corresponds directly to the file name identified when importing results (see section 2.2), and associates all the information within VEDA related to each run.

Time Period: correspond to the multiple time periods for which the model is run. Each time period is usually named by a representative year within the period (e.g. '2010' may represent the five-year period from 2008 to 2012). Besides the usual time periods, the Discounted Present value (DPV) (pseudo) time period is also defined. It is useful only when used with cost attributes, and allows the discounted present value of a stream of period costs to be shown.

Time Slice: sub-divide the periods according to seasons and time-of-day. In MARKAL and SAGE there are 6 time slices for electricity and 3 for low temperature heat. In TIMES, time slices are user defined and may vary depending upon the nature of the study.

Vintage: For selected technologies where technology characteristic change over time the vintage year corresponds to the year in which the technology was initially installed, as opposed to the column year which corresponds to the year in which the activity occurs⁶.

⁶ For MARKAL and SAGE this is only permitted for demand devices (and for some attributes only), whereas for TIMES any process may be vintaged, if desired.

Data Value: Primal or Dual, relevant only for optimization based models. In the MARKAL family of models, Primal, the default, is associated with all outputs corresponding to primal model variables (e.g. capacity, activity, energy flows, etc). The primal information may also include input and calculated values in addition to raw model results. The dual information consists of the shadow prices (dual values) of constraints (e.g. commodity prices)

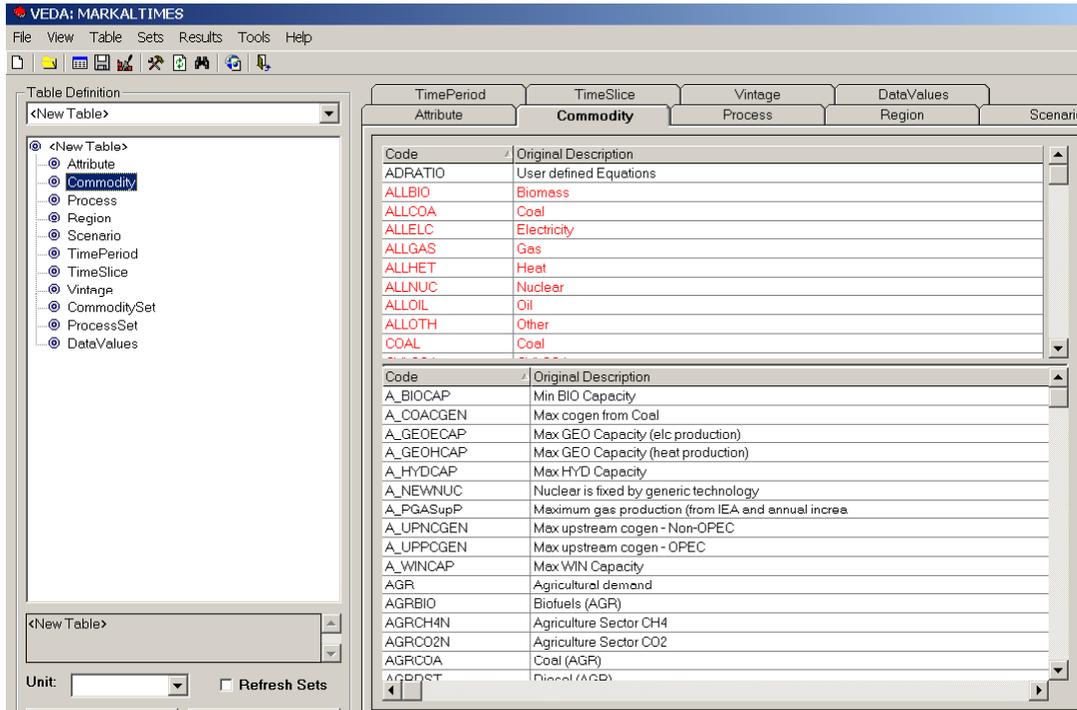


Figure 4. Main screen with the Commodity dimension selected

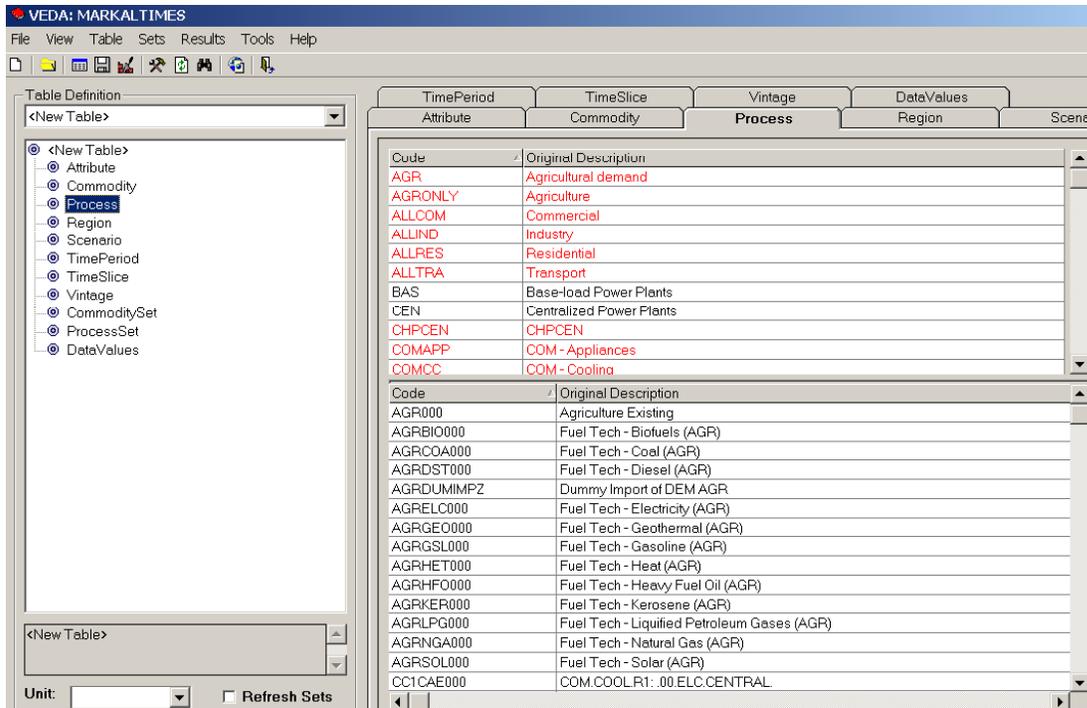


Figure 5. Main screen with the Process dimension selected

2. **The table layout window** appears when the corresponding button (at bottom-left on the screen) is pressed. It allows the user to change the default layout set earlier (see section 2.3) by dragging-and-dropping some dimensions onto rows or columns of the table to redefine the table's layout. The use of this window is explained in section 2.4.3.
3. The **Aggregation rules window** appears when the corresponding button is pressed (bottom left). This window allows the user to define specific arithmetic operations to be performed on retrieved information before being put into the table (see section 3)
4. The **units** window (bottom-right) allows the user to specify the units that will be associated with the table.
5. The **View table(s)** button processes the table definition, and produces the tabular cube view (see section 2.5) according to the elaborated criteria.

We are now in a position to define a new table.

2.4.2 Defining the contents of a new table

This is the main step in the creation of a new table. Creating a new table requires selecting which types of results to show, for which processes, commodities, regions, scenarios, time-periods, etc. This is accomplished as follows:

1. Click the down arrow of the top-left window to scroll through the list of table definitions and select "New Table" (or use the shortcut Ctrl+n).

2. Set *filters* on dimensions: In the MARKAL family of models, there are 9 dimensions, as discussed before. Most dimensions contain several and sometimes many elements, although some may be reduced to a single element, depending on the model runs (for example, the Region dimension may have a single element if the model is not multi-regional). Table 2 contains a list of MARKAL/TIMES/SAGE dimensions explicitly recognized by VEDA.

Note that the Dual solution value is only useful when the EQ_ComBal or EQ_Peak attribute has been chosen. In such a case, the dual value is the shadow price of the corresponding constraint (either a commodity balance constraint or a peak constraint)

Note on VEDA Sets: (sets will be discussed in detail in section 3): For the MARKAL models two of the dimensions, *Commodity* and *Process*, have been designed as indexes for which Set management is available. This results in the two dimension variants, called *Commodity Set* and *Process Set* respectively. Apart from these two, the other dimensions correspond to an index used by the various parameters, variables, and equations of the model. Defining sets may be very convenient for several reasons: first, the number of individual processes or commodities is very large, and thus their direct handling in a menu may in some cases be unwieldy, especially if hundreds of technologies or commodities are required for a particular table. In such cases, defining sets of commodities or processes may be much easier. Second, the naming of processes or commodities may well be different from region to region⁷, whereas the naming of sets of technologies or commodities is uniform (e.g. the set of fossil/nuclear/renewable electricity generating processes in MARKAL/SAGE are assigned to the sets FOS/NUC/REN, respectively, irrespective of the names of its members). Lastly, as will be seen in what follows, even though sets of technologies are used in a table, their individual members will still appear as individual rows or columns in the table, if desired. VEDA distinguishes sets defined passed from the model (listed in black) from sets defined by the user (listed in red). Methods to create new user-defined sets, or to alter previously defined sets, will be described in section 3. The original sets passed from the model itself may not be altered by the user in VEDA.

Filtering: A typical table examines only a fraction of the potentially huge amount of information that would be obtained if all elements of all dimensions were selected. Therefore, the user defines a table simply by limiting (filtering) the elements along each of the dimensions. This is done by selecting, for each dimension, one or more elements (or sets) for that dimension. To do so, click on the desired dimension in the left-side Table Definition Window or on the dimension tab on the window on the right. This will display in the right window the list of elements (and sets of elements in the case where the dimension is supported by set management) available for this dimension. Then, you may manually choose and delete elements in each dimension as follows:

- A. **Manual filtering:** Select the desired dimension tab in the right window. Then choose an element in the right-window, and *double-click* on it (or press the *space bar*, or press the *Insert key*) to transfer the element to the left-side table definition window. The chosen element is now displayed under the dimension heading in the window on the

⁷ In SAGE, a special effort is made to uniformly name the model's elements throughout regions. This however may not be the case for other models.

left. This operation can be reversed by highlighting the selected element in the left window and *double clicking* on it (or pressing the *space bar*, or press the *Delete* key).

A few remarks apply:

- If no filter (list of sets and/or elements) is applied to a dimension, all available elements will be selected for that particular dimension. The **only** exception to this rule is for the value dimension, where the default is the primal or calculated value. Also, note that for any dimension for which sets are maintained, the default is to ignore the set information (leaving the dimension out of the resulting analysis table) when no filter is provided..
- Many of the choices made at this phase of the table definition may be later altered in the View mode (section 2.5).
- When VEDA builds an analysis table according to a table definition, there is a quality control feature which checks for the existence of all the elements explicitly specified in a dimension. For example, if you define a table with scenario “S1” and delete the scenario from the database later, when this table is viewed again, there will be a warning “Scenario S1 is missing. Do you wish to continue?” To remedy this situation for subsequent requests, simply remove “S1” from the table definition.

B. Filtering via the Search Engine: in the case of dimensions containing many elements (e.g. commodities or processes in the MARKAL family of models), manual filtering may quickly become very awkward and time-consuming, due to the sheer number of elements in these two dimensions. There are two ways around this difficulty: the first one is to use sets (see section 3), if set management is available for that dimension. The second is to use an especially designed *search engine* to filter elements. This tool allows filtering via a combination of inclusions and exclusions based on the names (short or long) of the elements in the dimension being considered, or on their other characteristics. We use an example to illustrate the use of the search engine.

Example of use of the Search Engine:

We wish to construct a table of capacities of the processes used in the agricultural sector, excepting those processes that use coal as input fuel. With the process dimension selected as the current tab, we invoke the search engine by clicking on the *binocular* icon next to the process item list, which pops up the search engine window shown in figure 6. We now *select* all processes whose code *starts* with the string *AGR* and *exclude* all processes whose code *contains* the string *COA anywhere*. The bottom part of the window shows the results of the search, which may now be confirmed by clicking on the *Select all* button, or further modified manually if necessary, before pressing the *OK* button. The desired codes should now be highlighted in the element list, and should appear in the table definition window, as if they had been individually selected.

Remarks on the search engine:

- You may use the search engine either with individual elements or with sets, by choosing the associated *binocular* icon.
- If there are already some processes listed in the table definition main window, when using the search engine these will be **superseded** by the ones resulting from the search engine (future VEDA versions will provide an option to *add* elements rather than supersede them).
- To take full advantage of the power of the search engine the model builder should ideally establish careful naming conventions for elements, and include important keywords in descriptions (e.g., “Existing” for all currently installed processes)
- The search engine is very useful when the elements are numerous but otherwise obey simple naming rules. However, the resulting list is always defined in the context of a single table; therefore, if a particular list of elements is likely to be used in several tables, it may be more convenient to define a permanent set rather than a list. See section 3 for set creation.

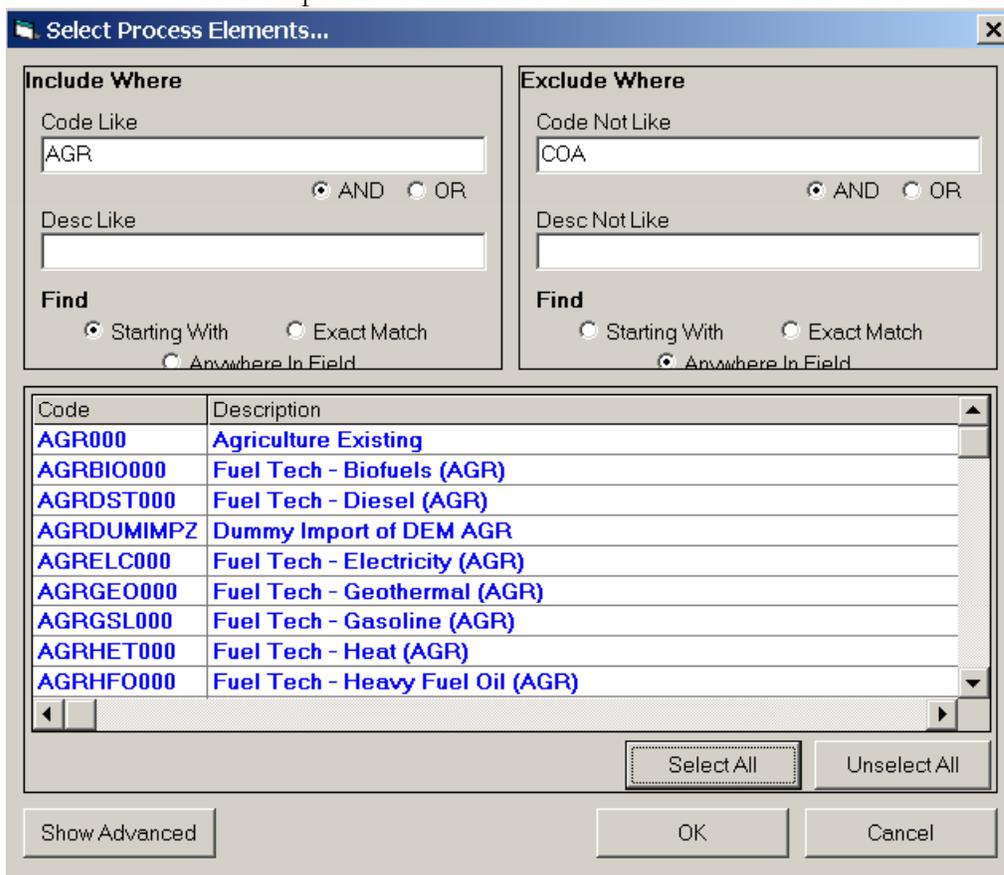


Figure 6. Invoking and using the Search Engine for processes

Global Filtering

It may be useful to globally apply certain filters and later be able to remove them for all tables, without having to redefine the tables. For instance, tables may have been defined for all regions and all time periods, but, for some portion of the report, the user may want to have the same tables for a subset of the regions and a subset of the time periods. To do so, while in the Region tab, right-click and select Global Filter. This will pop up a window to help you define the filtering of regions. Do the same for time periods. Then, to activate the just defined global filtering, simply check the Global Filter box in the main screen. To remove the filtering, simply uncheck the box.

2.4.3 Specifying the table layout

Click on the Bottom-left Layout button to open the layout window. The user specifies the layout of the table by deciding if each dimension should appear as rows or as columns of the table, or not at all. If a dimension is hidden (i.e. does not appear as either rows or columns), the table values will be summed up across all elements of the omitted dimension. Hidden dimensions are said to be '*on the page*', whereas shown dimensions are '*on the table*'. The user can select a maximum of 5 rows and 5 columns in this window (however, any number of rows and columns can be configured in the cube view -- see section 2.5). The fact that several dimensions may appear as rows (or as columns) of the table, implies that the user must also specify the *nesting order* of rows (or columns), i.e. which row dimension appears in the outermost position, which in the second outermost position, etc, until the innermost position (similarly with columns). The nesting order is selected by dragging-and-dropping each dimension to its desired position. When the cursor is placed over a dimension a crosshair appears. Then, holding the left mouse button down and sliding to a new position a green line will appear indicating that the dimension may now be dropped there. Any dimension not positioned as part of the row/column table layout definition appears at the top of the page. As already noted (and discussed further in section 2.5) these dimensions will have their values summed up. For each dimension thus 'on the page', if more than 1 value exists for that dimension, its name will be displayed between two asterisks (say, <region>).

Example (figure 7):

In figure 7, the *Scenario* dimension is selected as the outermost row label, the *Region* as the second outermost row label, the *Attribute* as the innermost row label. For the columns, the *Time period* was chosen as the only column label.

Additional layout options

The '*\Tools\Option*' menu in the main VEDA screen, allows the user to specify three characteristics of the table: under the *Dimensions* tab one may choose to show the element code and/or full description. Also on the *Dimensions* tab one may choose to subtotals along specified dimensions. Finally, the *General* tab allows the user to choose the number of decimals shown in table entries.

Remark: the table layout details defined via the `\Tools\Options` menu, are set for all future tables, and do not apply to existing tables. The settings hold until changed again. These defaults may be altered later in the *View* screen of VEDA (see section 2.5), for each individual table. The other six dimensions remain ‘on the page’.

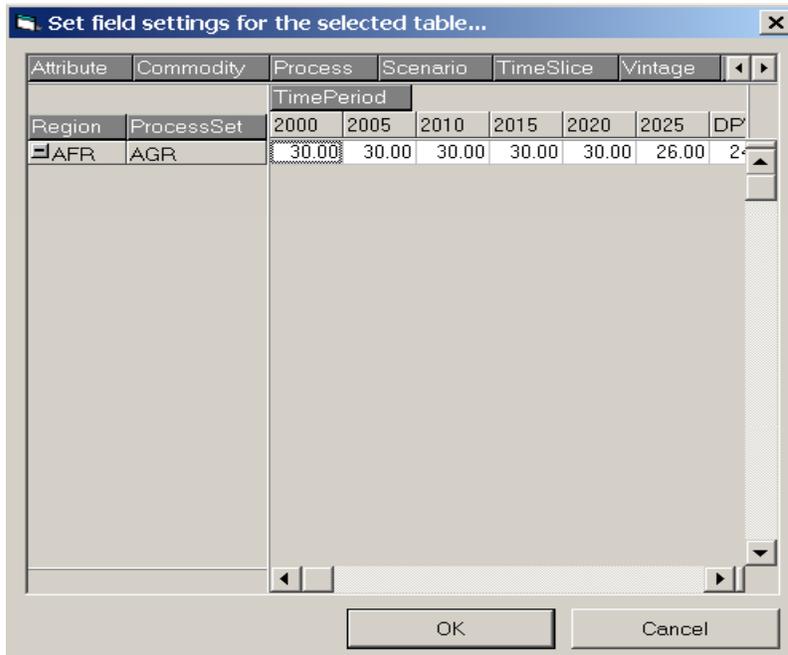


Figure 7. Defining the layout via the Layout window of the main screen

2.4.4 Aggregation rules

Aggregation rules allow for transformations to be applied to the various dimensions. The Aggregation button displays the Aggregation specification form, which is discussed in section 3.

2.4.5 Miscellaneous functions

DELETE Table

To delete several tables, use the main screen’s `\Table\Delete Tables` option. To delete the currently highlighted Table, press *Ctrl+D*.

SAVE Table

Ctrl-S saves the table definition, and prompts you to name the table, and optionally to add a longer description. The new table name will now appear in the list of saved table names in the table definition window. In VEDA, there are two opportunities for saving a table’s specifications: when constructing or changing a table’s specifications as just discussed, and again when exiting *View* mode (section 2.5), if the layout or content of the table view has been modified by the user. In either case, besides being asked if you want to save changes

made to existing tables, you'll also have the option of retaining or not the previous incarnation of the table by providing a new name for the just modified instance of the table.

EXPORT/IMPORT

A user may export a list of tables, which may then be imported by another user. To export a set of tables, choose the */Tables/Export* and fill the required information on the names of the folder and file to store the resulting text file, as well as the specification of the list of tables to be exported. To import a set of tables, use the */Table/Import* menu item and follow the same instructions.

2.5 Viewing a Table: the Cube view

Select the **View mode** by pressing the *View Table* button (if you have not saved the table yet, you will be prompted to do so now). VEDA then constructs the desired table applying the filtering criteria specified for the table. The View window is then shown, as illustrated in Figure 8. In this example showing the electricity production by type of centralized plants, the table has two levels of nested rows (Region and Process Set), and one column level (Time Period).

The screenshot shows the VEDA software interface with the following details:

- Table Name:** ELC_1_Central Production
- Table Description:** Electricity production from Central Power Plants
- Original Units:** PJ
- Active Unit:** Billion Kwh
- Data values filter:** /

Region	ProcessSet	2000	2005	2010	2015	2020	2025
AFR	Coal (ELC)	111.88	191.39	230.31	261.78	361.25	449.16
	ELC - NonHyd Ren	0.26	0.26	8.56	9.20	9.39	9.58
	GAS+OIL - mixed fuel (ELC)	~	0.00	0.00	0.00	0.00	0.00
	Hydro (ELC)	65.61	65.61	88.66	96.48	104.30	112.13
	Natural Gas (ELC)	0.00	0.00	40.66	75.64	15.84	15.37
	Oil (ELC)	169.39	162.61	157.73	153.00	148.41	143.95
	Uranium (ELC)	12.88	13.24	14.15	14.40	14.64	14.89
	Total		360.02	433.12	540.07	610.49	653.83
AUS	Coal (ELC)	161.03	171.47	182.71	193.11	212.53	259.74
	ELC - NonHyd Ren	3.39	8.63	8.63	8.63	22.35	22.42
	GAS+OIL - mixed fuel (ELC)	~	0.00	0.00	0.00	0.00	0.00
	Hydro (ELC)	40.64	41.09	42.11	43.14	44.19	45.26
	Natural Gas (ELC)	3.62	10.49	26.94	48.28	50.63	45.79
	Oil (ELC)	22.79	21.87	21.21	20.58	19.96	19.36
	Uranium (ELC)	~	0.00	0.00	0.00	0.00	0.00

Figure 8. View Screen for example (also called Cube view)

The View of the table is a fair rendition of how the table will look when printed. The table name is displayed in the top left section of the window, and the units just below that.

The View window has a tool bar with several icons. The name of each button is shown when the cursor is positioned over the corresponding button. Moving from left to right here is a brief description of the tool bar buttons:

- Decrease or Increase the number of digits after decimal;
- Save Changes (s). The first button saves the highlighted table only, and the next button saves all tables currently on screen (in the case of multiple tables). If changes were made while in View modes, the user is asked to provide a new name or to use the old name for the table. At this point, it is important to make a distinction between saving a table's specifications and saving the table's Cube file as well. This latter operation is triggered by the option `\Tools\Options\General`, and checking the box called '*Save Cube Files*', as discussed in section 2.3. If chosen this option will display an existing table almost instantly (as long as the content of the table has not changed since the last request --VEDA keeps track of any potential changes), whereas if not, the tables presented in the *View* mode are reconstructed each time they are requested via the *View Table* button.
- Sort the Table (three options: default, ascending, descending,);
- Hide/Show blanks/zeros, or filter elements to be shown according to their values; The filtering of values allows the user to show only values within certain ranges (for instance values whose absolute values are larger than 0.1). Note carefully that the filter is applied to values expressed in the original units of the table, even if the current units are different. When the filtering is applied, the entries that have been filtered out are either replaced by a ' ~ ' sign or are completely absent from the table, if an entire row or column is filtered out.
- Export table (four options: Excel, Word, Html, Text) or copy a portion of the table on to the clipboard; When the table is exported to a file, the date and time are appended to the table's name to constitute the name of the saved file.
- Produce a graph of a selected table area: first block the area, then press the graph button;
- Print and Preview the current table
- Set layout preferences (appearance, font, colors) as well as the way multiple windows are shown (cascade, horizontally, vertically). These options are useful when several tables are viewed simultaneously. Note that you may simultaneously view several tables by repeatedly pressing the View button in the main screen after highlighting the appropriate tables. This may be useful for comparing results from several tables.
- Close *View* mode (you may also close individual tables by clicking on the 'X' in the frame of that table's window).

- Press the  button to remove the header portion of the View screen and keep only the table itself.

As we shall see in the next few subsections, the View mode allows much flexibility and adds power to the table definition features described earlier. With View, you will be able to view the table you have just defined, and, more importantly, to review and revise the choices made regarding the contents and layout of a table, as well as alter the default choices regarding subtotals and labeling, and finally to save, export, and print a table.

All changes to the layout of a table performed in the View mode, and *most* (but not all) changes in contents, are remembered by VEDA and incorporated in the Table definition, where they become permanent. See the remark in section 2.5.2 for the exception to this rule.

The View shows the Table as you defined it via the Table definition screen (section 2.3) with the following small difference: if a certain dimension is not relevant, then it is not shown at all in the viewed table. For example in the MARKAL model if the level of the capacity variable is the only attribute included in the table, then the commodity set and commodity dimensions will not be shown in the table, since they are not relevant for capacity results. This provides for a cleaner, less cluttered presentation.

2.5.1 Altering table layout in the *View* mode

As already mentioned, most of the layout choices made when defining a table may be modified in the VIEW mode. It is often more convenient to use the VIEW mode to finalize the table's layout, since this is done with the table shown on the screen, allowing the user to better appreciate the appearance of the table.

Layout changes may be performed either by drag-and-drop or by right-clicking when the cursor is in a particular area of the View window, as was described in section 2.4.3, and repeated below for convenience.

Modifying the dimensions shown in the Table

For new tables the dimensions that were not specified by the user as rows or columns, either in the default table layout or when describing the table, are displayed at the top of the table (see figure 8) upon entering *View* mode. We will say that these dimensions are “**on the Page**”, as opposed to the shown dimensions, which are said to be “**on the Table**”. The user may move any dimension from its current location to another location chosen among: on page, as column, or as row. When dragging a dimension from one location to another, a cross-hair first appears, and with the left-mouse pressed a green bar will indicate valid drop positions on the table (row or column) or to the “on page” hidden area. This feature allows the user to alter the choices made earlier in the definition of the table contents and layout. Typical examples are: moving a dimension from a row to a column position, moving a dimension out of the table altogether (i.e. moving it to the page position), re-organizing the nesting order of the rows.

For each dimension “on page” position (i.e. not in columns or rows), the results are automatically summed over all its elements. Therefore, some dimensions should **never** be

put on the page if summing along this dimension makes no sense (for instance, the scenario dimension). If a dimension is on the page and has more than one element, its name appears preceded and followed by an asterisk ‘ * ’ to remind the user that results are summed up over these elements in the current view. In the example shown in Figure 8 this is the case for the *Process* and *Time Slice* dimensions.

Viewing descriptions and/or codes

1. Right-Click the mouse while the cursor lies in the desired row or column area.
2. Select Display Type from the menu.
3. Choose *Show Description* to obtain long names, or *Show Code*, if you wish to revert to short names, or *Show Both*, to show both codes and descriptions. The column is presented re-sorted according the description/code respectively.

If you right-click in the VIEW window but when not residing on a row or column text area, you will be offered a slightly different menu consisting of just the options for adjusting various attributes of the cube, dimension you wish to alter, plus graphing/exporting

As already discussed in section 2.3.1, the same result may be obtained by choosing the */Tools/Options/Dimensions* sequence in the main menu. This sets the default for all future tables, whereas when in VIEW the option applies only to the currently active table.

Computing Totals

We describe the case of row totals:

1. Right-Click while the cursor lies in a particular row text area.
2. Select or deselect *Totals*.

In the example of Figure 8, totals are shown for the Region dimension only.

Note that the same result can be obtained by choosing */Tools/Options/Dimensions* from the main menu on the main screen (section 2.3). Again, this will be used as default for all future tables, whereas doing it in the VIEW mode affects only the current table. If the table is then saved upon existing the total request will remain active when the table is requested again.

Collapsing

By clicking on the “-” sign to the left of every element, the user can collapse (sum up) all the values along the dimensions *coming after this row or column*. Note that such aggregation does not always make sense (for instance, what would the aggregation of two scenarios mean?). It is the user’s responsibility to only form aggregates that make sense. You will sometimes be helped by inverting the nesting order of rows so that the innermost rows may now be meaningfully aggregated.

To re-instate the full list of all elements simply click on the “+” sign display for collapsed rows or columns.

Hiding/Showing “null” rows and columns

Do this by clicking on the icon  shown at the top of the screen. This results in only presenting the rows for which a non-zero or non-blank entry appears in at least one column of the table. This feature is particularly useful when a table shows differences between two scenarios (see Aggregation Rules in section 3). In this case, many rows are likely to be blank, and hiding these rows for which both scenarios have the same values improves legibility, especially when com.

Other layout options

Several items in the tool bar allow the user to change the appearance of the table, its colors, fonts, etc. We leave the user to experiment with these various options.

2.5.2 Altering table contents: Removing elements from a dimension

Two methods are available:

Manual removal of elements

- Click on the white arrow beside a dimension. This will display the list of elements available for display. The elements with a check mark are currently “on the Table,” while those that are “de-selected” remain on “on the Page” but are not displayed.
- Select (or de-select) the elements to be included (or removed) from the current view. This feature allows the user to alter the choices made earlier in defining the table contents.

Note that there is no reminder that an element has been hidden.

Filtering out of elements:

A more powerful tool is also available for filtering out undesirable elements. If for instance you wish to apply a filter to elements in the *ProcessSet* dimension, proceed as follows (the same option is available for all dimensions):

- Put the cursor over the row or column where the process sets are displayed.
- Right click and select *Apply Filter*
- In the filter window that appears, define which elements should be filtered out, using the same methods that were used in the search engine described earlier.

Note that although you may remove some elements in any dimension, and reinstate them, you cannot *add* elements that were not in the dimension to start with! For instance, if you have initially selected the fossil fueled generation plants in the *process* dimension, you may remove (say) all coal plants, but you may not add (say) hydro-electric plants. To add new elements, you must go back to the Table Definition stage.

Important remark: If you remove elements from a dimension in the cube table, these elements will *remain removed* when you save the table again. However, there is an exception to this: if the table uses sets (of processes, commodities or attributes) and if you remove individual elements of such a set, the set will **not** be altered by your choice, and the next time you view the table, the eliminated elements will reappear. If you want to permanently alter the definition of the set, you will have to do so via the set editing process (section 3).

2.5.3 Exporting result tables

Data presented in the current View may be exported in various formats. Two methods are available:

First method

- Click on one of the *Export to Excel/Word/HTML/Text* icons at the top of the window. This operation will create a file of the requested type in a pre-specified directory (section 2.3).
- The file will be automatically named by appending the date and time at which the file was saved to the table's name.

Upon completing the Export operation VEDA will prompt the user to specify if the exported information should be presented using an appropriate application.

Second method

Another way to copy all or part of a table consists of right-clicking while the cursor lies in the numerical area of the table, and choosing the Export option. The table is then put in the Windows clipboard, and may then be pasted to an application of your choice. This manner of operating is often more convenient when debugging a run's results or if the tables must be pasted at specific locations in a workbook or document for further processing.

Both of these methods also allows to export a portion of the table, by first blocking the rectangular area to be exported, and then only right-clicking to choose the export option.

2.5.4 Graphs

A portion of a table may be conveniently graphed as described here.

- Block a rectangular area in the table
- Click on the Graph icon on top of the VIEW window. You will get a tentative graph, that you may now modify by swapping axes, switching to bar, line, pie, area, etc., as well as two or three dimensional charts.
- You may copy the graph to the clipboard for later printing or importing into a document.

2.5.5 Print

The table may also be printed directly from the View mode. The Print Preview button allows modifying the printing format (margins, orientation). In addition, printing font and font size may be modified by using the appropriate buttons in the tool bar.

3 ADVANCED FEATURES

3.1 The batch mode

Result analysis often involves two distinct modes. One is exploratory in nature, where the analyst designs new result tables to dig into model details useful for debugging or to suit a particular report, or export a few selected results from a new run for further processing. There is a second mode, which may be called “production mode.” Here the table definitions and layouts have been stabilized, and it is desired to produce a set of tables and graphs without further user interference. To address these two different needs, VEDA has two modes of operation: **interactive** and **batch**. This choice is available under the */View/Execution Mode* option. All we have seen till now about VEDA has been in the interactive mode of operation.

If you choose the batch mode, the table definition window of the main screen is replaced by a list of existing tables, each preceded by a check box. The user may then check any number of boxes, and VEDA will produce them as a batch.

The following batch mode related 3 options may be selected from the */Tools/Options/General* tab menu:

View Cube: check the box if you wish to see the tables displayed on screen. In batch mode, the View shows different tables on different tabs to facilitate browsing and modifications.

Export: check the box if you wish to have the tables exported to a file on the disk. If you choose to export, then specify the format to export to, and the location of the exported files. You have the option of exporting all tables in the batch to the same spreadsheet or to different sheets of a workbook (one table per sheet)

Print cube: will print the tables.

With the Batch feature and options the user can easily reconstruct standard sets of Tables into a complete Report as runs are refined and new runs performed.

3.2 User sets

When first using VEDA-BE with a model, the only sets initially available are those passed from the model by the GDX2VEDA utility. For the MARKAL family of models these correspond to the standard ones defined for processes and commodities (e.g. all electric plants (ELE), resource supply options (SRCENCP), transformation processes (PRC), and several others). In most cases, the user will want to define various other sets to help in the analysis of results. For example, if it is important to see the installed capacity of all fossil fired electricity generation technologies, the FOS set already exists in MARKAL, whereas if gas fired generation technologies have to be examined separately, a new set may be defined

that includes only these technologies. In VEDA, set creation and modification has received particular attention, because of the potentially large number of elements associated with some dimension, for example processes in a MARKAL model, often numbering in the hundreds or even the thousands, for each region. Due to this volume, the analyst requires a powerful yet flexible and accurate tool to facilitate the creation and use of complex tables.

3.2.1 Creating a new set

The set creation feature is accessed via the */Sets /NewSets* main menu path, which pops up the window shown in figure 9. When working with a MARKAL model, after choosing *Commodity Set* or *Process Set*, the new set is constructed using inclusion rules, and/or exclusion rules acting on elements and/or previously defined sets. For any given model, the dimensions for which set management has been deemed necessary are all available for defining user sets.

The user may define a new set by first using one or more already defined sets, and/or putting filters on the names (either codes or descriptions), and/or directly selecting individual elements. When specifying the names of elements to be included or excluded, the wild cards “?” and “*” may be used with their usual meaning (‘?’ being a single character wildcard, and ‘*’ being a general wildcard for any number of characters embedded in the name). It is not too important to be very precise in the first attempt at defining a new set, because the user will have the possibility to manually delete (but not add!) set members subsequently.

Figure 9 shows the creation of a commodity set for the Africa region, containing all commodities from the set **EFS (Fossil energy carriers)**, whose code starts with the ‘**AGR**’ string and has six characters, but does not end with the string **COA** (hence, the set thus constructed contains the agricultural sector fossil fuels with the exception of coal).

Once the set is specified, press the *Load Parent Dimension* button to view the qualifying elements on the right portion of the screen. You may now manually delete some undesired elements from the list by unchecking them. Once the set membership is satisfactorily defined, press the *Create Set* button and a window opens to let you name the set (up to 10 characters) and provide a description. You also have the option of approving the same set definition for all parent dimensions (regions, in the case of MARKAL models) by pressing the *For All Parents* button (if that option is not selected, you will have to define the set separately for each region).

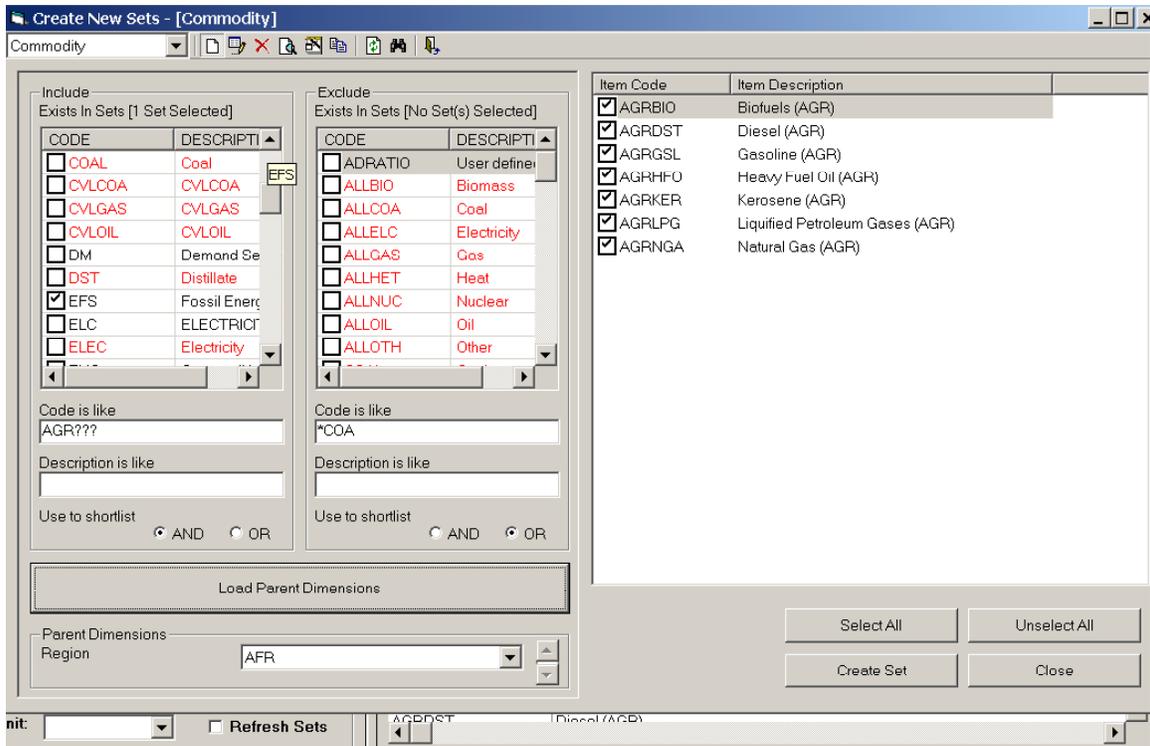


Figure 9. Process Set Creation screen

Two remarks apply:

- The user should not click the *Create Set* button when the Parent Dimension (Regions for MARKAL) box is empty, as the name dialog box will not be permitted to complete (and if For All Parents is hit this may cause a fatal error).
- The set's *name* is chosen only once, and is the same for all regions. This is important, as you need to have a common set name for all regions, when defining the table dimensions.

Several additional features add power and flexibility to the construction of sets.

The inclusion and exclusion conditions can both be chosen. If both are chosen, the conditions are applied sequentially, starting with the inclusion conditions. For example, one may easily create the set of renewable non hydroelectric generation technologies by choosing *'exists in REN'* and *'does not exist in HDE'*.

It is possible to declare multiple elements in the *name is like/not like* boxes, separated by commas. These commas are interpreted as logical **OR**. You may also select more than one existing set in the *Exists in Sets* (or *Does not Exist in Sets*) lists, by ticking several boxes from these lists. This is equivalent to specifying the union of these sets.

For the MARKAL family of models special support is provided for Process sets permitting them to be defined by their inputs and/or outputs: in the *Create New Process Set* screen, you may choose to define the set by its inputs and/or its outputs by using the appropriate options. For instance, the set of all technologies consuming geothermal heat (GEO) may easily be constructed as shown in figure 10. The resulting set has five elements (processes) in Africa.

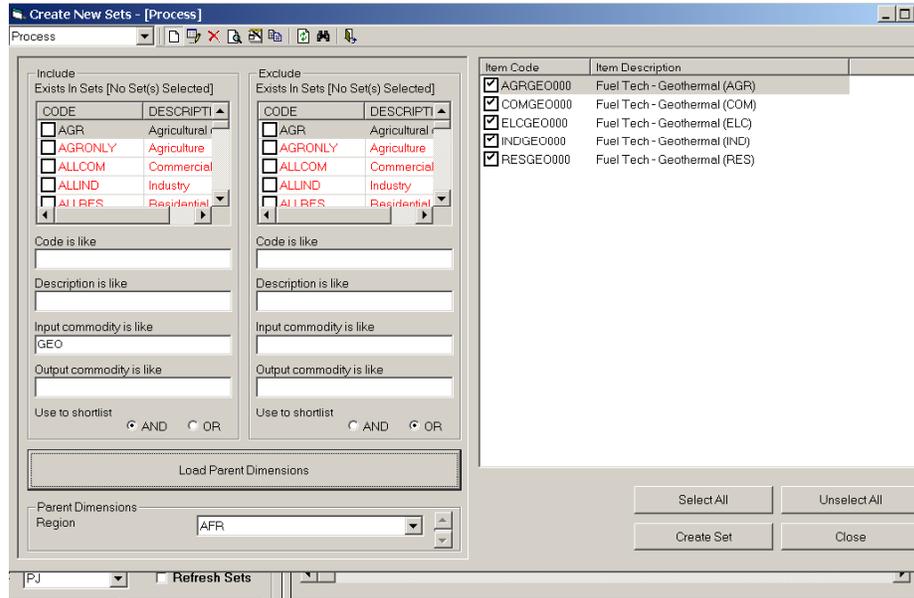


Figure 10. Construction a set of processes that consume geothermal heat.

The user may also use the *binoculars filter* specification tools (by hitting the symbol at the top of the New Set form) to select master sets and collection of elements. The rules for using the *binocular filter* are the same as those of the previously described search engine.

Dynamic set creation

Another powerful set creation feature consists of creating a process set or a commodity set based on information shown in the current View.

For instance, you may want to create a commodity set containing all commodities shown as rows of a table of results. To do so:

- Make sure the commodity rows are on the table (i.e. not on the page);
- Position the cursor on the commodities rows, make a right click, and choose the option '*Create Commodity Set*,' and
- You will be shown the usual *Set Creation* window, but without any provision for adding elements: all you are permitted to do is to choose a set name, and manually delete elements before saving it by pressing the '*Create Set*' button for each region.

This feature is very useful when it is desired to create a set of all commodities involved in a particular sector or subsector. Similarly, you may want to create a process set containing (say) all processes that produce a certain commodity.

There is a subtle but real difference between process set creation via inputs or outputs (as described in the last bullet of the previous subsection) and dynamic process set creation: the former is based on the RES, whereas the latter is based on results from a particular set of runs.

3.2.2 Set Updating (Refresh)

VEDA remembers the *set of rules* used to create each user-defined set. If at a later time the model data base is altered, resulting in alterations of the list of elements in some dimensions, VEDA is capable of updating the definition of user-defined sets by choosing the */Sets/Refresh sets* option of the main menu. The same option may also be activated from the top row of the *Create Set* screen. This important feature insures that the user will not have to redefine sets he had defined earlier, as long as the rules he used to define those sets are still adequate for the new database. For example, in the above described example, if a later version of the model adds one fossil fuel in the agricultural sector, say AGRLFO, refreshing the set will automatically add the new fuel to the already defined set, without having to go through the set editing process.

It is important to note that such updating of previously defined user sets takes place automatically when Importing results (see section 2.2).

3.2.3 Exporting and Importing sets

It is also possible to *import set definitions* from other users. To do so, one user may write the sets information to a text file (using the */Sets/Export set(s)* menu item), which can then be imported by another user (using the */Sets/Import set(s)* menu item). Choosing these menu items opens a sequence of windows to specify the names and folders for storing/retrieving the sets, as well as listing which sets to export or import.

3.2.4 Viewing and editing user defined sets

You may browse an already defined user set by choosing the */Sets/Examine Sets* menu item.

You may alter a set definition by redefining its contents for some or all regions. To do so, use the */Sets/Edit Sets* main menu option. You will get a set editing window such as the one shown in figure 11. If you press the *Edit* button, you will be in a position to alter the set formation rules, just as if you were creating the set again. The redefined set will supersede the old one.

An alternate access to set editing or viewing is available from the Table specification form: when a tab for which set management is supported (e.g., the process or commodity tab for MARKAL) is selected and some particular set is highlighted, right-click and the *edit* option is selected. You will then be directed to the exact same set editing window just described.

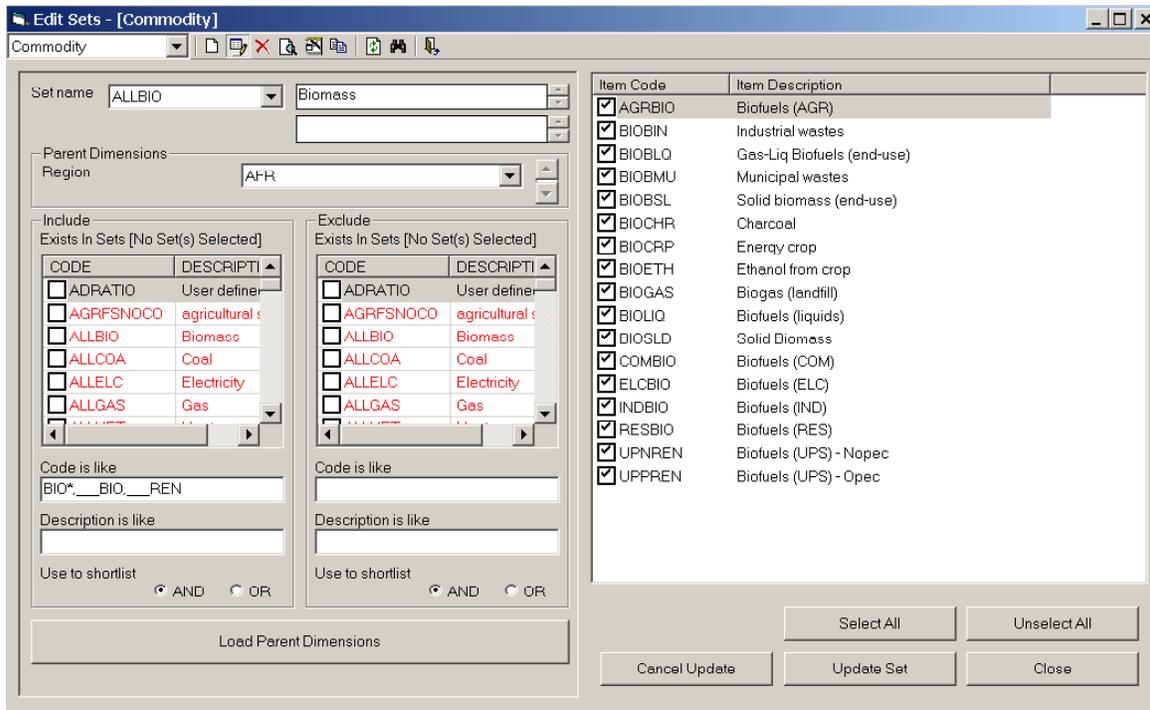


Figure 11. Edit Set window showing the current set creation rules.

3.3 Aggregation Rules

In some cases, it is desirable to build a table containing not only the raw data or results, but some processed version of them. For instance, one may want to construct a table that shows the difference between the Baseline scenario and an alternate scenario. Another case arises when one wants to perform simple calculations based upon the values of various data presented in the View (e.g., show the effective utilization factor of each generating equipment (i.e. the ratio of production over capacity of each ELE technology)). Each of these cases, and others like them, involves some arithmetic operation on attributes. The *Aggregation* button at left-bottom of the main VEDA screen allows these operations to be defined and their results to be included in a Table.

There are 6 types of aggregation operations: *Collapse*, *Difference*, *Product*, *Ratio*, *Enumerate*, and *Expression Builder*. We describe below three of them: Expression Builder, Collapse, and Enumerate. The *Difference* operation is only permitted when the user explicitly selects just 2 elements for the dimension for which the difference operation is to be performed, with the (2nd – 1st) being calculated and included in the View. The others are self-explanatory.

In addition, there are a few cases where the order in which aggregation operations are done is important. The default priority order for each dimension is set initially at 0, and may be modified by putting the appropriate number in the square box next to each dimension. Priority 1 is highest and priority 10 lowest.

3.3.1 Expression Builder and Collapse operations

Figure 12 shows the Aggregation window for the Utilization factors example mentioned above: in the sample results, capacity is given in GW, and the production is in PJ. Proceed as follows to specify how VEDA is to compute the utilization factor:

Make appropriate choices under all dimensions. You should have chosen VAR_FOUT (output from processes) and VAR_CAP (installed capacity) under *attribute*, when defining the table contents. The process set is ELE, and the commodity is ELC

Had the units been consistent, we could have simply chosen these two attributes in the above order, and chosen “*Ratio*” under aggregation rules for the attribute dimension. However, since we also have to divide by 31.536 (to harmonize units) we need to use the general *expression builder*:

Therefore, choose “*Expression Builder*” as the aggregation rules for the *Attribute* dimension and, using the drag-and-drop calculator on the right portion of the window, construct the expression “[VAR_FOUT]/[VAR_CAP]/[31.536]”.

Note that VEDA uses the so-called *Reverse Polish* priority order for arithmetic computations, i.e. the operations are conducted strictly from left to right, and no parentheses are used. For instance, the mathematical expression (A-B)/C is written as follows in VEDA: [A]-[B]/[C]

You may also choose a label to name the expression you have just built. For example, VAR_FOUT/VAR_CAP can be called UTILIZATION FACTOR. Type this name in the box just below the expression box and it will be available when viewing the table.

Important Remark: The two VEDA attributes we want to divide have different indexes. Specifically, VAR_FOUT is time-sliced (that is divided into 6 segments) while VAR_CAP is annual (just one segment). Also, VAR_FOUT has a commodity dimension, whereas VAR_CAP does not. In order to work with such variables in an expression, the inconsistent dimensions need to be “collapsed”. In other words, the variables will be aggregated along these dimensions. To do so, ‘*Collapse*’ should be chosen as the aggregation rule for *TimeSlice* and for *Commodity*. Furthermore, an order must be specified for the two operations (collapse and expression builder). We want *collapse* to occur first, and *expression builder* second. Therefore, we choose 1 as priority order for *Collapse* and 2 for *Expression builder*.

The user has the choice to hide the original elements in the table display (and only show the results corresponding to the expression – the utilization factor, in our example). To do so, check or uncheck the box labeled *Hide Original* that appears next to the attribute. If that box is not checked, both the original attributes and the expression are shown initially in the Cube view.

Press OK to save the Aggregation Rule, then save and view the table.

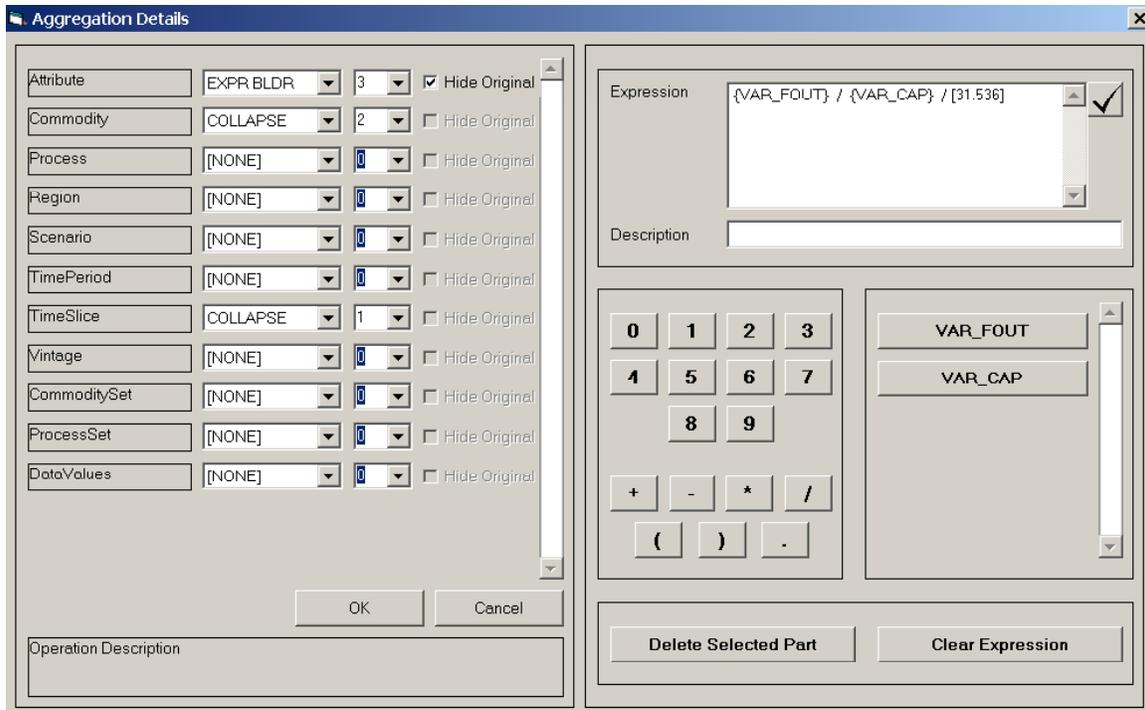


Figure 12. The Aggregation window with Expression Builder example

3.3.2 Enumerate

This option allows creating a compound table. If you choose *Enumerate* for a dimension (e.g. the Region dimension), separate sub-tables will be created for each instance of that dimension (e.g., one table for each region). Note that you may use *Enumerate* for up to three dimensions. The suffix *x* on *ENUMx* indicates the nesting order when more than one dimension is enumerated.

In the View mode, each sub-table will be shown on a different tab, to facilitate browsing and altering the table. As already mentioned in section 2.5, some operations (such as display and export) may be performed separately on each tab of an enumerated table or on all tabs simultaneously.

3.4 Viewing the Reference Energy System (for the MARKAL family of models)

This feature constitutes a convenient tool for exploring the Reference Energy System (RES) of the modeled region(s), but also for examining the results of a scenario, at a very detailed level and in the direct context of the RES. It is an excellent debugging tool. The basic idea of this function is to draw a diagram of a portion of the RES centered at a particular process or commodity, and to show the values of the various flows of energy and materials in and out of a process, as well as the primal values of the process variables such as capacity and activity and the dual values (prices) of the commodities. The user may navigate downstream or upstream of the current RES view, and thus trace processes and commodities as far backward or forward as wanted from the point of initial access to the RES.

This function is accessed by selecting */Results on RES/Show Results* from the main menu. [The options is disabled for non-MARKAL applications.] A screen such as that of Figure 13 appears, but without any information in its main window. The screen has a top menu, several boxes, and a main window (currently empty) which is destined to show the RES and results centered on a commodity or on a process.

In order to choose the commodity or process at the center of the shown RES portion, proceed as follows:

- Press the arrow next to the *binocular* button and choose *process* or *commodity*. Then, from the pop-up window, use the appropriate search methods to select a *single* process or a *single* commodity. Highlight this single element and press *OK*;
- Select one scenario, one region, one time period, and one time-slice (if applicable), and
- Press the upper leftmost button [RES icon] to refresh the screen.

Description of the information on screen – Centering on a Commodity (figure 13)

The center box shows the name of the commodity (ELC, electricity) and its shadow price. At left, each process producing ELC is listed, as well as the amount of ELC it produces. Further left, each commodity entering each process is identified. At right of the center box, each process consuming ELC is named with the amount of ELC it consumes. Further right, each commodity produced by each process is designated by its name.

Description of the information on screen - Centering on a Process (figure 14)

In this case, the central box contains the name of the selected process (*ELCCOA000*). At its left are shown the commodities consumed by this process and at its right the commodities produced by it. Below the RES picture a window shows the names, prices, and amounts of these inputs and outputs (for instance, in figure 13, 400.2 PJ of Brown Coal are consumed and its shadow price is equal to 0.8 \$/GJ).

Navigating through the RES

Using the above method one may observe any portion of the RES. To move through the network click on a commodity or a process shown in the current RES view. The next RES view will be centered on the selected process or commodity. Thus, figure 15 shows the screen obtained after clicking on the Brown Coal commodity name in the screen of figure 14.

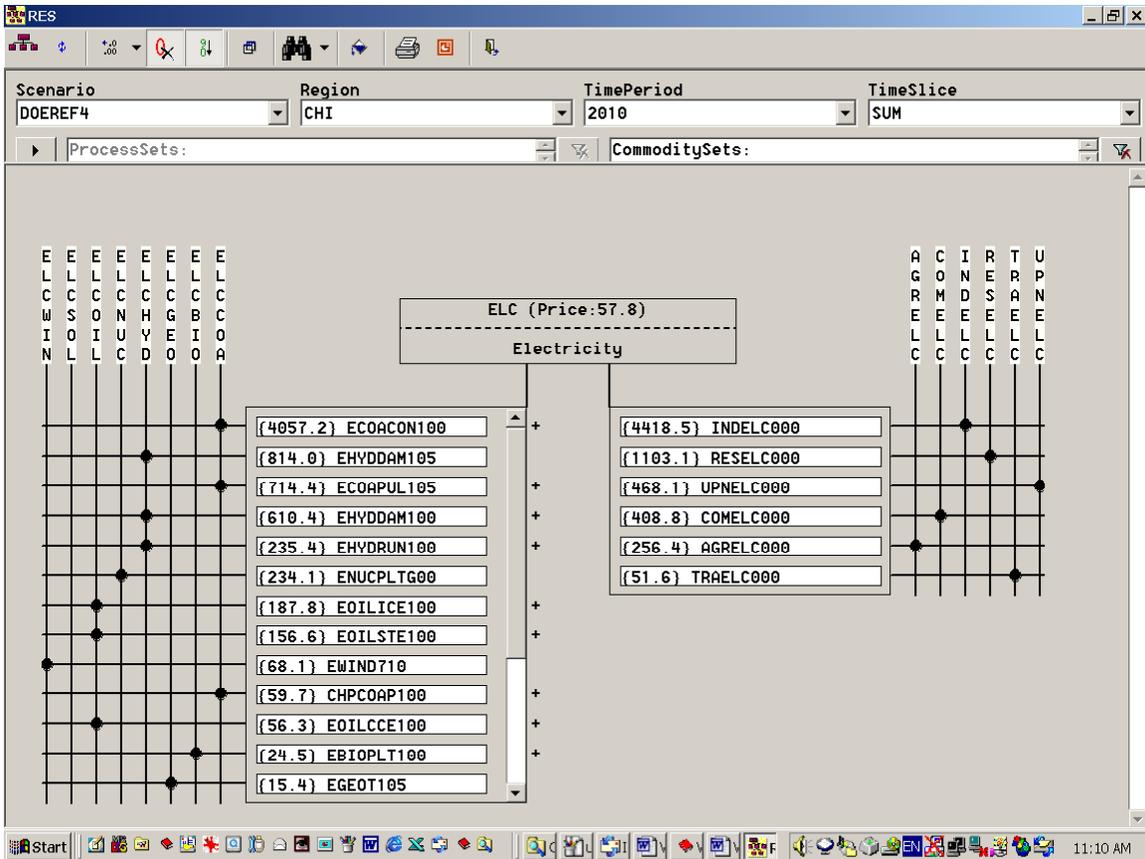


Figure 13. RES screen centered on the ELC commodity

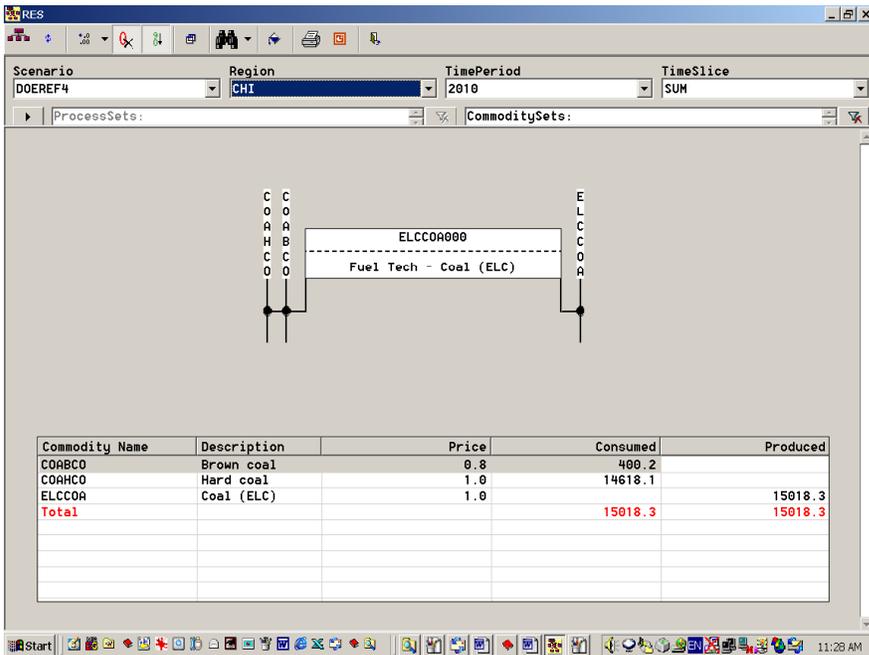


Figure 14. RES screen centered on the ELCCOA000 process

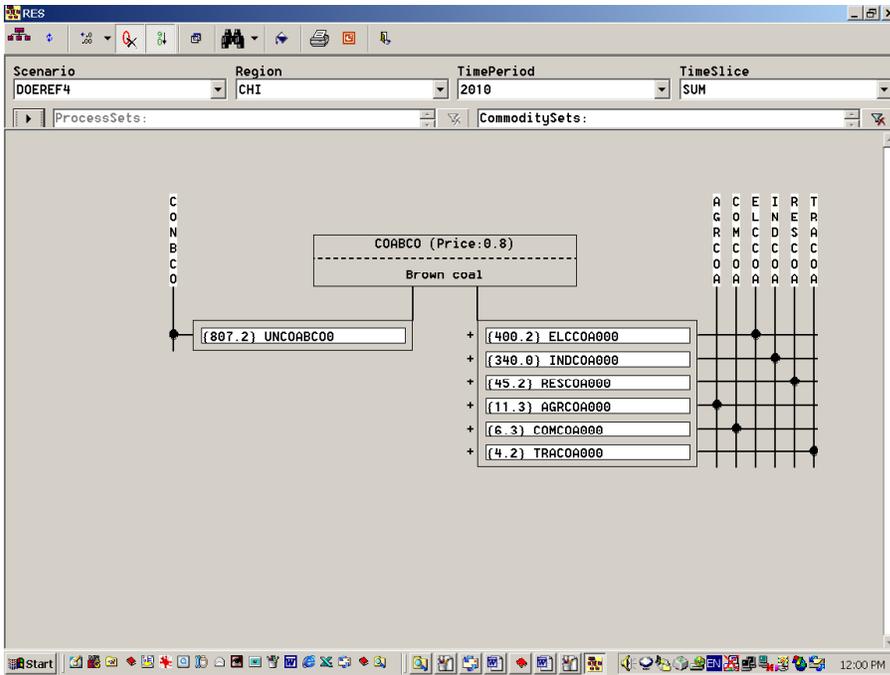


Figure 15. RES screen centered on the COABCO commodity

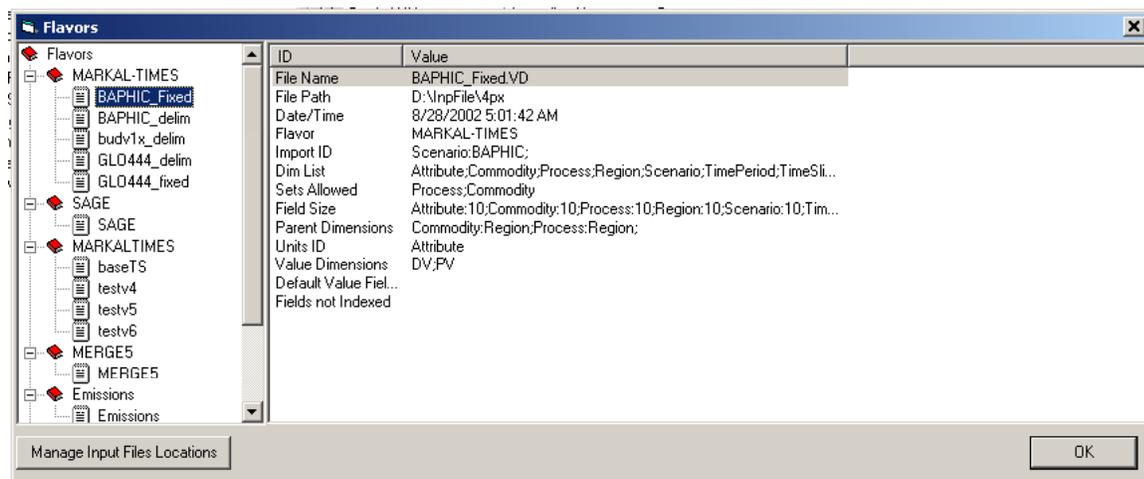
4 Reference Section (Main Screen Menu)

4.1 File Menu

4.1.1 New database⁸

The user can create a new VEDA database in a folder of his choice. A new database is created using the header information found in the <application_case>.VD data file as created by GDX2VEDA.

1. A dialog showing a list of <case>.VD files available in the input files paths declared by the user is displayed.



2. Alternatively, if no <case>.VD file is found in these paths, the user is prompted to either add some relevant paths, by means of the Manage Input Files Locations button, or to cancel the operation.
3. Select a <case>.VD file that corresponds to the application of interest and click OK to create the database.
4. The new database is activated immediately after VEDA completes the creation task, and the <case>.VD file from which it was created is imported into the database.

⁸ The New Database option is NOT available for all Flavors of VEDA, as some flavors are keyed to a specific model and its associated VD header structure.

Note that when Importing Results the only <case>.VD files listed, regardless of the paths in the *Manage Input File Locations* list, are those with VEDA headers that exactly match that of the current open database.

4.1.2 Open database

User can switch between several databases, and thereby models or applications, according to his need. This is made possible thru the “Open Database” menu.

1. A dialog showing a list of folders on the is displayed similar to Windows Explorer.
2. Select the path in which the database to be chosen resides.

VEDA performs a series of operations when switching databases that might take some time. Also, the user must make sure that no other VEDA operation is midway when using this feature.

4.1.3 Compact database [F12]

Compacting the VEDA database that is currently open is at times desirable to ensure top performance, or as a first order attempt to fix “erratic” behavior by VEDA. It is of vital importance that there should be no interaction with the database while compacting. This might result in corrupting the database.

4.1.4 Backup database

Backup allows the user to save a copy of the VEDA database that is currently open. User is required to provide the name for the current backup. The backup file is kept in the same folder as the original database path.

4.1.5 Restore database

A list of available backups is displayed. The user has to select the backup name he wants to restore. After restoration, the backup still remains in the folder.

4.2 View menu

4.2.1 Toolbar

Visible: shows/hides the toolbar on the main form.

Large icons: shows large icons on the toolbar.

Small icons: shows small icons on the toolbar.

4.2.2 Statusbar: Toggles Show/hide the status bar at the bottom of the main form.

4.2.3 Open folder(s): Activates VEDA Table folders.

- A dialog is opened with a list of available folders.
- The user can select any number of folders.
- Clicking “OK” activates the selected folders.

4.2.4 Show/Hide Tabs

With this feature, the user can control the dimensions displayed as the tabs on the right side of the main table specification form.

A list of dimensions is displayed with the previously visible ones as selected and rest as deselected.

The user can select any dimensions.

Clicking “OK” would hide the unselected dimensions and display the selected ones.

4.2.5 Reference Energy System (Only for the MARKAL Family of models)

Invokes the RES subsystem discussed in detail in section 3.4.

4.2.6 Execution Mode

Interactive mode: work with individual tables

Batch mode: work with collection of tables organized into reports

4.2.7 Search [F3]

Displays a context-sensitive search dialog form depending upon the active position on the main form.

1. Tables – activates the table(s) selected.
2. Units – activates the selected unit in the unit combo.
3. Dimensions – selects the elements selected for the current table.

4.2.8 Refresh [F5]

Refreshes the main form by:

1. Reloading tables lists;
2. Reloading units lists;
3. Reloading dimensions lists, and
4. Resetting menus and form setup.

4.3 Table menu

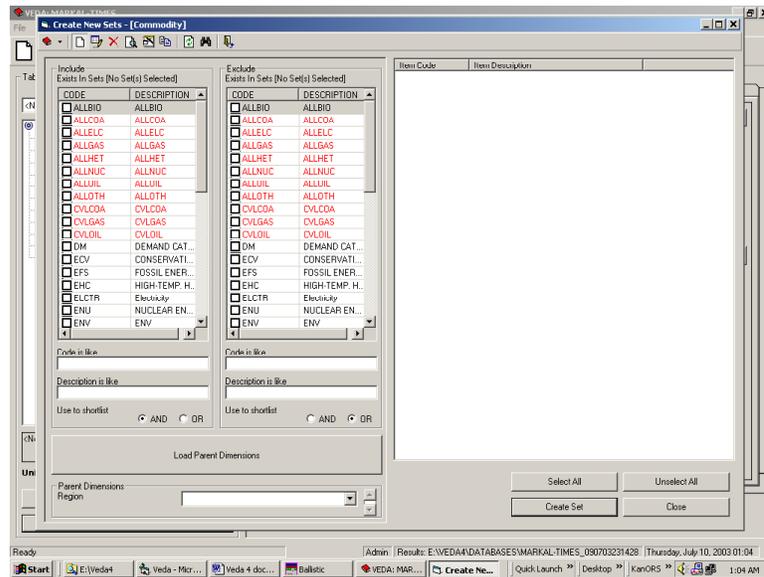
- 4.3.1 New Table [Ctrl+N]
- 4.3.2 Save Table [Ctrl+S]
- 4.3.3 View Table [Ctrl+L]
- 4.3.4 Delete Table [Ctrl+D]
- 4.3.5 Rename Table [Ctrl+F2]
- 4.3.6 Import Table(s)
- 4.3.7 Export Table(s)
- 4.3.8 Table Master [Ctrl+M]
- 4.3.9 Delete Cube File(s)
- 4.3.10 Sort Tables
 - Ascending
 - Descending

4.4 Sets menu

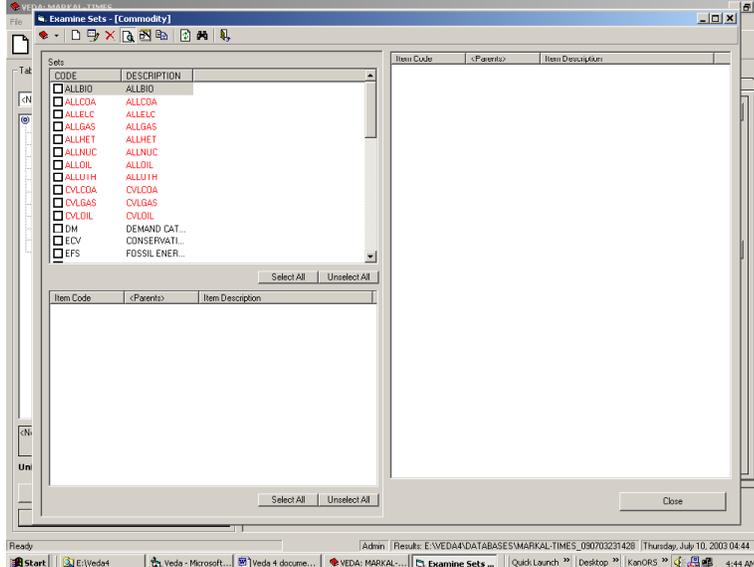
- 4.4.1 New Set(s) [F6]

This feature activates a new screen with several options for selecting elements to form a new user-defined set.

1. Include – Specify selection criteria. All the criteria are optional.
 - a. Exists in sets – select the pre-created sets that include the elements you want to include in the new set.
 - b. Code is like – pattern string to match the element code.
 - c. Description is like – pattern string to match the element description
 - d. Use to shortlist – option to search by “AND”ing or “OR”ing the various selection criteria defined. Default is “And”.
2. Exclude – specifies de-selection criteria. All the criteria are optional.
 - a. Exists in sets – select the pre-created sets which include the elements you do not want to include in the new set.
 - b. Code is like – pattern string to match the element code.
 - c. Description is like – pattern string to match the element description.
 - d. Use to shortlist – option to search by “AND”ing or “OR”ing the various selection criteria defined. Default is “OR”.



3. Load Parents/Shortlist button – click to shortlist elements based on the criteria specified.
4. Parent dimensions – if parents are allowed for the dimension, separate lists for all the parents are loaded when the “Load Parents/Shortlist” button is clicked. The first valid combination of parents will be made available to the user by default.
5. Elements list – the short-listed elements are listed in this list. It is not necessary for the user to create a set by using all the elements listed here. One can deselect elements here also.
6. Select All/Unselect All buttons – select/unselect all elements in the results list.
7. Create set button – initiate create set task using the selected elements. A small dialog is displayed in the middle of the screen prompting for the set name, description (optional), and a set comment (optional). Clicking OK will create the set for the selected parent combination. Clicking “For All parents” on this final form will create the set in all the parent combinations.
8. Close button – closes the form



1. Select the sets from the sets list, on the top left corner, that are to be deleted fully or for only selected parent dimensions.
2. The parent combinations for each set, that are available, are listed in the list in the bottom left. These are all selected by default.
3. Also, the elements of the parent combinations selected are listed in the elements list on the right side.
4. Click “Delete Set” button to delete.

4.4.4 Examine (View) Set

Makes the set elements visible to the user for the selected sets. Similar to delete, the user has the ability to view for all the parent combinations, or for a few selected ones. But, unlike delete, the user can view the system sets (imported thru IARD) sets also.

1. Select the sets from the sets list, on the top left corner, that are to be displayed.
2. The parent combinations for each set, that are available, are listed in the list in the bottom left. These are all selected by default.
3. The elements of the parent combinations selected are listed in the elements list on the right side.

4.4.5 Refresh Sets [Ctrl+F5]

Facility to auto update sets based on the preset set definition after Import operations. For example a new data set is imported into the database, and certain sets need to be refreshed. So, instead of editing each set, or creating new sets, one can use this feature to update all the sets at once.

4.4.6 Import Set(s)

Facility to bring user-defined sets into this database which were created in another database (for this application).

1. Follow the dialog box to point VEDA to the file containing the Exported sets definitions. Select the file.

2. A form listing all the previously Exported user-defined sets contained in the file is presented.
3. Select the sets of interest, or all, and hit OK to initiate the Import activity.
4. If any of the sets to be imported already reside in the current database the user is prompted to override or not. If Yes then any existing set is replaced, if No said sets remain as they are at the time of the Import request.

4.4.7 Export Set(s)

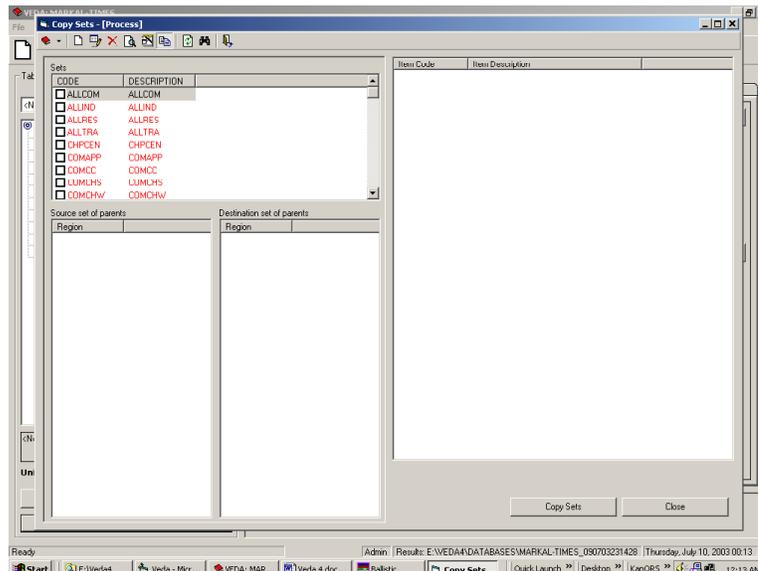
Facility to move user-defined sets to a file in preparation for Importing into another database (for this application).

1. Follow the dialog box to point VEDA to the folder into which the sets are to be exported, and name the file.
2. A form listing all the ser-defined sets is presented.
3. Select the sets of interest, or all, and hit OK to initiate the Export activity.

4.4.8 Copy Across Parents

A set definition is copied from one set of parents to one or more other such sets. It is to be noted that only the definition is copied here, and if the elements are originally available for the parents, only then they are included in the set. Only, user-defined sets can be copied.

This feature is available only when a dimension allowing sets and having parents is included in the model flavor.



1. Select the sets to copy from the sets list. This will load the parent lists.

2. Select the source combination of parents for the sets to be copied. Note that all the sets selected must be available in the source parents.
3. Select the set of parents to which the set must be copied. If the selected set(s) already exist in a destination parent, the destination parent will be marked RED.
4. When you are ready with your selections, click Copy.

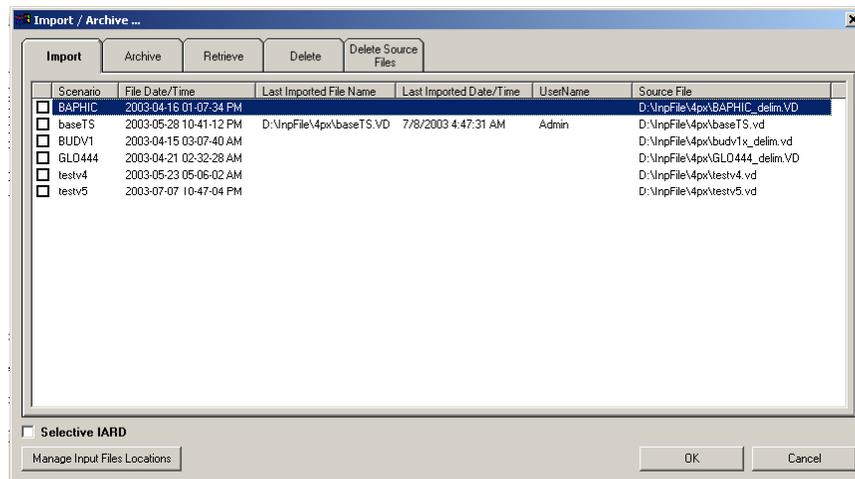
4.5 Results menu

4.5.1 Import/Archive... [F7]

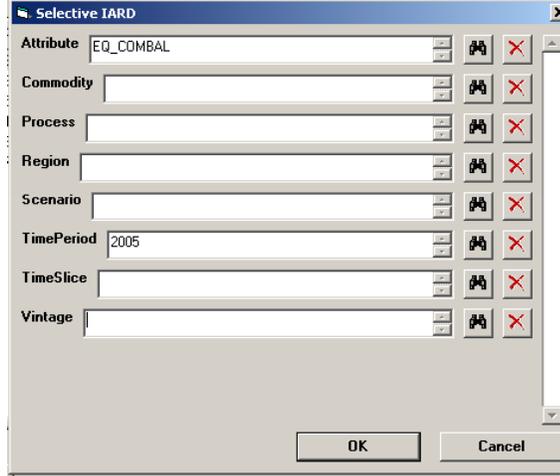
The Import/Archive/Retrieve/Delete (IARD) engine is used for data input and dataset management in VEDA. Using these options, one can import various <case>.VD files in a single database, given they are of same flavor and dimensions, as well as manage them both in the database and in the application folders.

1. /Results/“Import/Archive” menu or “F7”.
2. According to the tab, an appropriate dialog window is displayed listing
 - a. Import – all the <case>.VD data files in all input files paths (established by means of the Manage Input Files Location button) which correspond to the structure of the current database are presented. The form also shows and when and by whom they were imported, if they were.

Note that if Import is requested for an already loaded dataset, all information associated with that dataset are first deleted, or may be removed to archive, prior to the actual loading of the new information.



- b. Archive – all the ImportIDs (corresponding directly to the original file names, cases for MARKAL) that had been imported into the database are available to be archived. Archived datasets are dumped data that is not used

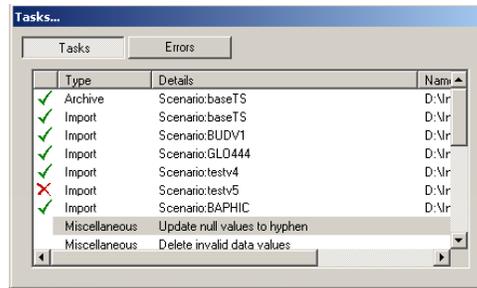


for analysis purposes, but the user can conveniently recall it for use when needed.

- c. Retrieve – all the Archived datasets are available to be retrieved. Once retrieve they are treated like any other active online dataset, and including in all queries, if appropriate.
 - d. Delete – all the datasets, whether imported, archived or retrieved can be deleted. Once deleted the only way to re-establish a dataset in VEDA is to import it.
 - e. Delete Source Files – the <case>.VD files in all input file paths specified are listed. Selecting them for deletion removes the set of <case> files (the .VD, .VDE, .VDS, .VDT) from the specified location.
3. Depending upon the tab all or some of these action buttons may be available.
 - a. Selective IARD – Allows the user to import only a part of the <case>.VD information from the file.
 - b. Manage Input Files Locations – Opens the input files manager window permitting the user to add additional paths to the list of folders to be checked for compatible <case>.VD files. Available only with the Import tab.
 - c. Delete All – Deletes all the IARD datasets (except those archived) from the database. Available only with the delete tab.
 4. Click OK to proceed with the IARD operation
 - a. If a standard Import has been requested a progress box displays the status of the various tasks that VEDA must perform to carryout the import process.
 - b. If “Selective IARD” is checked, a dialog is displayed. Here one has to input the selection criteria for some or all dimensions.

After the selection criteria, if any, have been taken care of, a dialog listing various tasks to be performed is displayed and the IARD engine starts processing the tasks one by one.

- i. The tasks that failed to complete are marked by a cross (✘) and the successful ones by a check (✔).



5. After all the tasks are complete, a list of errors which caused the tasks to fail, if any had, are listed. Use “Close” or “Esc key” to close the error dialog.

4.5.2 Import Topology Data [for MARKAL models only]

Allows for importing of <case>.VDT files with the RES topology information separate from the IARD operation.

1. A list showing all the <case>.VDT files in the input files paths is displayed.
2. Select the files to be imported and click “OK”
3. Topology data in the selected files is imported.

4.5.3 Select Database

This feature enables a user to use select a different database than the one currently used. The user has to select the folder in which the new results database is contained. This can also be accomplished by means of the /File/Open Database operation.

4.6 Tools menu

4.6.1 Reset Cube Settings

An option to reset the various color, font, and other settings made to the cube by the user. It is to be noted that this option only works on the display settings for the VEDA *View* data cube.

Accessing this menu will change the following settings –

Row Alignment	to Left Top
Row Style	to Raised Light
Column Alignment	to Left Top
Column Style	to Raised Light
Cursor Style	to Light (Thin Border)
Grid Style	to Flat (Solid Lines)

Show/Hide Page Fields	to Show
Data Not Available Caption	to “”
Data Fore Color	to Black
Back Color	to White
Data Font	to “MS Sans Serif”, 8
Grid Color	to Light Gray
Printer Grid Color	to Black
Headings Back Color	to White
Headings Fore Color	to Black
Headings Font	to “MS Sans Serif”, 8
Fields Back Color	to Dark Gray
Fields Fore Color	to Black
Fields Font	to “MS Sans Serif”, 8
Totals Back Color	to White
Totals Fore Color	to Black
Totals Font	to “MS Sans Serif”, 8
Printer Font	to “MS Sans Serif”, 7.5
Printer Headings Font	to “MS Sans Serif”, 7.5
Printer Totals Font	to “MS Sans Serif”, 7.5

4.6.2 Set Default Layout

Helps apply the default layout for dimensions in the VEDA *View* data cubes. These settings are applied to all new tables, if different layout settings are not applied explicitly.

1. A small data cube is displayed with all the fields at their appropriate places, i.e., the places where they were set previously. If this is the first time this setting is being made, all the fields appear in the page area.
2. The user can shift the fields according to his requirements using drag-and-drop.
3. Click “OK” to apply the settings, or “Cancel” to abort the task.

4.6.3 Delete All Cube Files [Shift+Del]

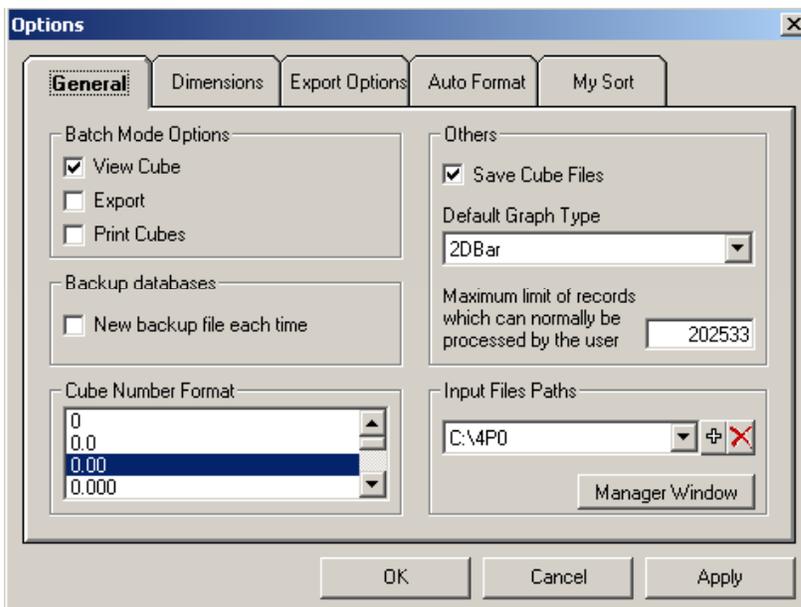
Deletes all the saved .CUB files from “..\Cube Files\<<Flavor>” folder.

4.6.4 Options [Ctrl+O]

General Tab

1. Batch mode options

- a. **View cube** – show data when VEDA tables are opened in batch mode.
 - b. **Export** – export data to applications selected in the export options tab when VEDA tables are opened in batch mode.
 - c. **Print cubes** – print data when VEDA tables are opened in batch mode.
- 2. Database backup options**
- a. **Auto backup** – backs up databases, before all the important operations like IARD, sets import, table import, etc.
 - b. **New backup file each time** – A new backup file is created each time the user backups the databases manually. This option is not available to “Auto Backups”.
- 3. Cube number format option** – Number format that will be applied by default to the VEDA tables created by the user henceforth.
- 4. Input files paths** – Paths from which the <case>.VD files are picked for operations like IARD and New database. One can add a new path using “+”, and delete existing using “X”. A more interactive interface is the “Manager Window”.
- 5. Other options**
- a. **Save cube files** – Option to save cube files (*.CUB) in the “..\Cube Files” folder for VEDA tables being viewed. Saving cubes can dramatically speed the task of constructing repeatedly requested tables.
 - b. **Graph type** – Default type of graph, which will open when the user views a graph for the VEDA tables data.
 - c. **Max limit of records** – An upper limit of records that can be generated for VEDA tables. Above this limit, the user will be prompted to process the table or not. At that time the user will also be prompted to raise the limit if desired.

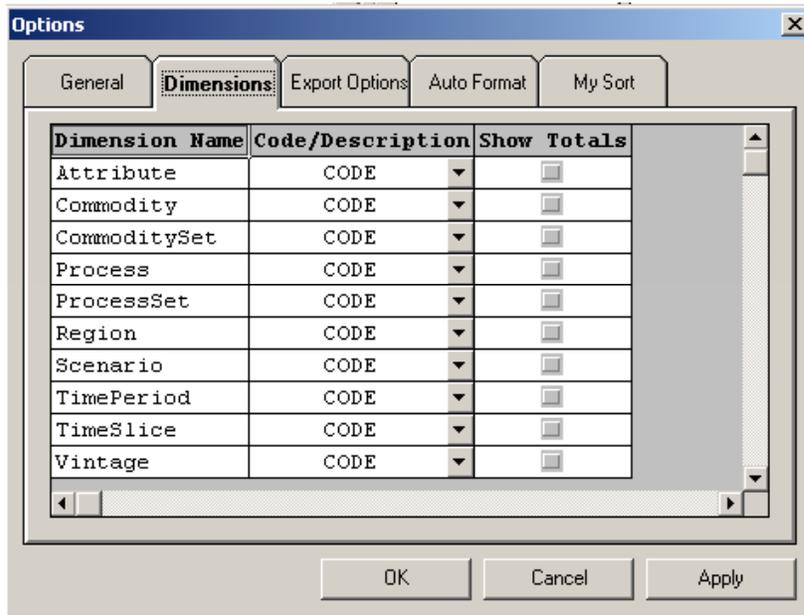


Dimensions Tab

Allows the user to set the default layout for new VEDA tables created henceforth.

The “**Code/Description**” setting defines what is displayed for the dimension in the data headers: Item code, Item description, or both.

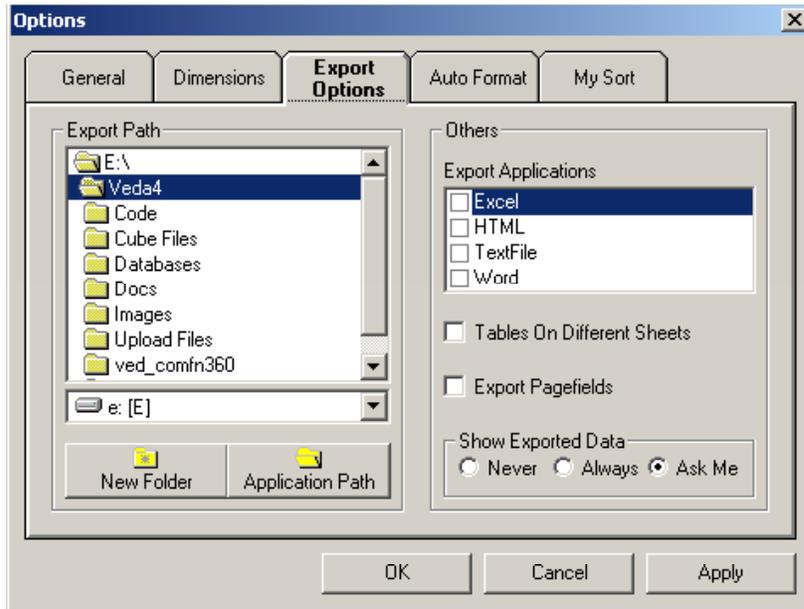
The “**Show Totals**” setting indicates where totals are to be loaded for that dimension in the data cube.



Export Options Tab

1. **Export Path** – Folder where the data exported from VEDA tables will be kept. Note that the initial default is the root of the C:\ drive.
2. **Others**
 - a. **Applications** – Data may be exported in four formats: Excel, Word, HTML, and Text. Selecting some of these formats in the Export Application box, will activate the corresponding exports when the Batch mode has been selected, and only if the Export box is checked in the Tools/Options/General tab.
 - b. **Table on different sheets** – option for Excel export; specifies if all the VEDA tables are to be exported to the same worksheet in the export destination workbook, or separate worksheets are to be created for each table.
 - c. **Export pagefields** – export pagefields data from the VEDA table or not.

- d. **Show exported data** – displays data after export or not. This option is available only for exports from Interactive mode.



Auto Format Tab

The user can apply various format settings to the exported data, like font, fore color, fill color, etc. these settings can be applied by selecting the areas on the grid and right clicking to make selections.

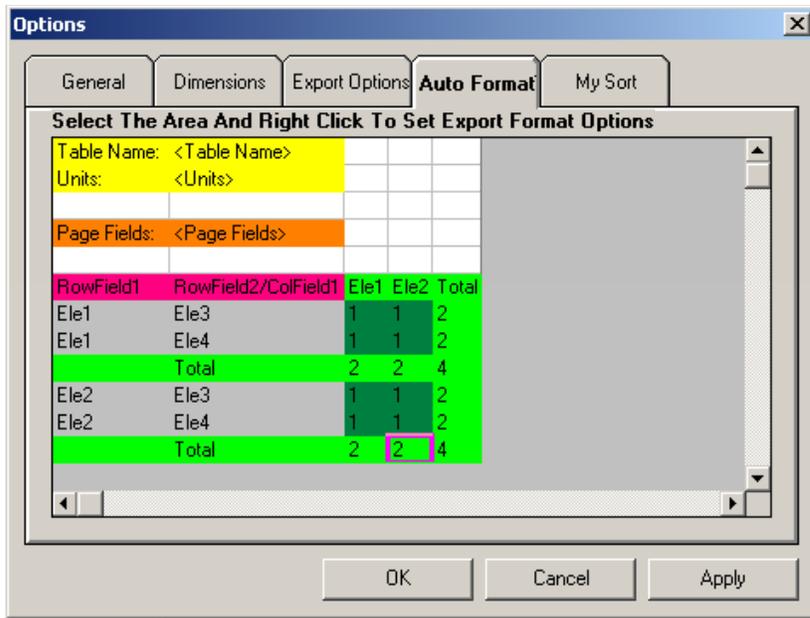
The various areas available are –

- a. Table name
- b. Units
- c. Page fields area
- d. Dimension names area
- e. Row headers area
- f. Column headers area
- g. Data area
- h. Totals area

Settings that can be made are –

- a. Font name, size, color, underline, bold, italics, strikethrough
- b. Fill color
- c. Borders – not available to Table name, units, pagefields areas.

It is to be noted that these settings will not be applied to the text files.



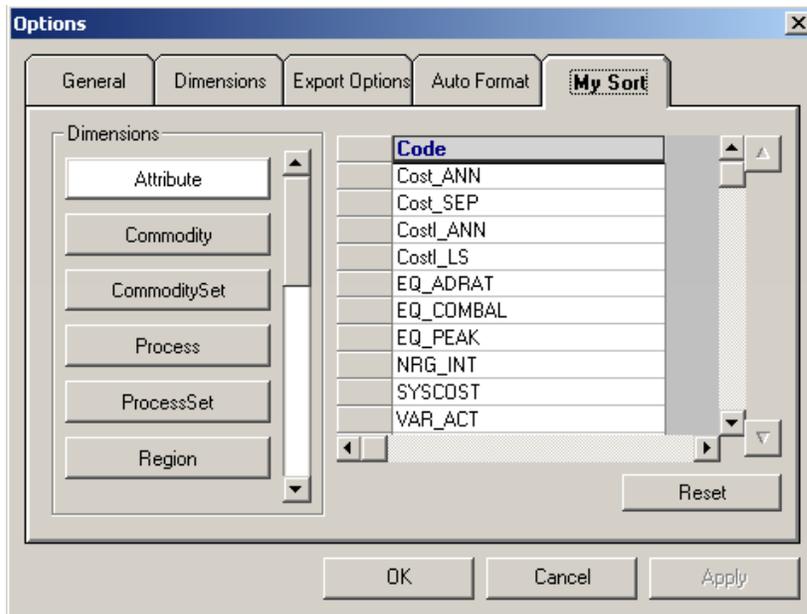
My Sort Tab

VEDA permits the user to create a custom sort order for the dimension elements to be displayed in the data cube.

The user can switch between dimensions by clicking the dimension tabs to the left. One can set the order by moving the elements up or down by using the Up/Down buttons on the right side of the elements list.

This option is useful when the elements of certain dimensions such as regions, scenarios, etc must be shown in specific order (different from alphanumeric).

Once this option has been declared for a dimension, *My Sort* will appear as an option when right-clicking on that dimension in a Cube table. In order to remove the My Sort option, press the *Reset* button in the *Tools/Options/My Sort* tab



4.6.5 Units

Create/Edit/Delete

Export unit(s)

Import unit(s)

4.6.6 Reset database [Ctrl+F12]

4.6.7 License Operations

Update License

Write to file

4.7 Help menu

4.7.1 About Veda

4.7.2 Contents [F1]

4.7.3 Search [Ctrl+F1]

4.7.4 KanORS.com [Shift+Ctrl+F1]

Appendix A: GDX2VEDA Data Definition File

The GDX2VEDA GAMS utility allows anyone familiar with a particular model to specify what information is to be extracted from the GAMS model (any parameter, variable or equation) and passed to VEDA. In this Appendix we first simply reproduce the GDX2VEDA --help echoed Help File. It explains the way to call GDX2VEDA, as well as the various options available to the user in the VEDA Data Definition (VDD) file.

For those working with the MARKAL family of models an appropriate initial VDD file is distributed, which requests the dumping of the information summarized in *Appendix B: VEDA MARKAL/SAGE/TIMES Attributes*. However, with a proper understanding of the VDD file, after saving the standard VDD, a knowledgeable user may carefully augment the VDD to request additional information be moved from GAMS to VEDA.

{Note: This section of the Appendix will be redone as the VDD is finalized.}

```
gdx2veda gdx vdd [run]
```

```
gdx  GAMS GDX file
vdd  VEDA Data Definition file
run  VEDA Run identifier (optional)
```

The VEDA data file name and run identifier are either taken from the gdx file name or specified with the run name. Use "token with blanks" if needed.

```
>gdx2veda mygdx    // will dump the gdx symbols
>gdx2veda          // prints this message
>gdx2veda --help  // prints more detailed help message (this!)
```

Add .csv to the run name to write the VEDA dump files in csv format.

VDD File Summary

```
[DataBaseName]9
myveda
[Dimensions]10
tab_name short_name
tab_name
[DataEntries]11
veda_attribute gams_name name1 name2 ...
veda_attribute gams_name
[DimensionText]12
tab_name gams_set
[DimensionTextAll]13
tab_name gams_set
```

⁹ Name of the application.

¹⁰ The various tabs to appear on the VEDA screen.

¹¹ The individual data structures (Attributes) to be managed by VEDA.

¹² Optional list of individual elements of the tab_names with their descriptions. Otherwise, all instances found in the data are enumerated and their associated description passed to VEDA. Once 1 provided the data generated list is ignored.

¹³ Optional list of additional elements of the tab_names not found in the set/index in the model data but whose description is desired.

```
[SubSets]14
  tab_name sub_name gams_name
-----
```

<myveda> is usually the application name which will be displayed on the top of the VEDA splash screen. When a new VEDA database is created, a new folder with this name will appear: ...veda\database\myveda_date_time.

Where date and time are the creation time stamp.

<tab_name> corresponds to the tabs of your VEDA screen
 The <short_name> is optional, if omitted, the <Tab_name> is used. The use of a <short_names> is convenient because <name>s are usually long. The GAMS indexes are often good candidates. The name "Attribute" is reserved to indicate VEDA keys.

<veda_attribute> is the name of the data items appearing on the Attribute tab in VEDA. The gams_name corresponds to the parameter or variable/equation (.L/M for level/marginal), and name 1-n are the ordered tab_name or short_name corresponding to the actual index positions in the model component.

Lines starting with * and empty lines are ignored. Blanks, commas and tabs are delimiters, blanks before and after delimiters are ignored. Quotes around data items are optional. The input data is NOT case sensitive.

Example of a VEDA Data Definition file

```
* Transport model
```

```
[DataBaseName]
  TrnPort
```

```
[Dimensions]
* VEDA tab-name   GAMS short_index
Attribute
Plants           i
Warehouses       j
```

```
[DataEntries]
* veda_attribute gams_name tab1 tab2 ... for gams index 1, 2, ...
"x(i,j) duals"  x.m           i Warehouse
Shipments      x.l           i j
SupplyPrice    supply.m      i
DemandPrice    demand.m     j
TransportCost  c             i j
Distance       d             i j
Supply         a             i
Demand         b             j
TotalCost      z.l
SupplyNodes    i             i
DemandNodes    j             j
Rate           f
```

```
[DimensionText]
```

¹⁴ For each Tab for which set management is to be performed by VEDA the system sets and subsets need to be provide.

```

* gams_set tab
  i i
  j j

[SubSets]
* sub_name gams_name tab
  il ic Plants
  il id i

```

Example of a TIMES Data Definition file

```

* TIMES GDX2VEDA Data Definition File

[DataBaseName]
TIMES

[Dimensions]
All_Tech p
Attribute attr
Commodity c
Region r
Period t
Vintage v
TimeSlice s

[DataEntries]
* VEDA Attr GAMS - indexes -
*** Variables
  VAR_Act var_act.l r t v p s
  VAR_ActM var_act.m r t v p s
  VAR_Cap var_cap.l r t p
  VAR_CapM var_cap.m r t p
  VAR_FIn f_in r v t p c s
  VAR_FOut f_out r v t p c s
  VAR_Ncap var_ncap.l r t p
  VAR_NcapM var_ncap.m r t p
*** Equations
  EQG_Combal eqg_combal.l r t c s
  EQG_CombalM eqg_combal.m r t c s
  EQE_Combal eqe_combal.l r t c s
  EQE_CombalM eqe_combal.m r t c s
  EQ_Peak eq_peak.l r t c s
  EQ_PeakM eq_peak.M r t c s
*** Costs
  Cost_AnnInv vda_aic r v t p
  Cost_AnnO&M vda_aoc r v t p
  Cost_RSupply vda_rsc r t p c s v
  ObjZ ObjZ.l

[DimensionText]
* Gams_set_name Veda_Tab dummy_name

[SubSets]
* subset GAMS VEDA Tab

```

Appendix B: VEDA MARKAL/SAGE/TIMES Attributes

Attribute Name	Meaning	Involves dimensions *
Cost_ANN	Annual costs calculated as the sum of fixed and variable operating costs plus fuel delivery costs	Process, commodity (for delivery cost only)
Cost_SEP	Cost of supplying a resource (annual)	Source, commodity
Cost_TRD	Cost or revenue from trade, calculated at exporter prices	Commodity, region1, region2
CostI_AN	Investment cost in annualized form: the investment cost is assumed to be amortized over the life of the process	Process
CostI_LS	Investment cost in lump sum (period cost)	Process
EQ_ADRAT	The level and marginal value associated with user-defined (ADRATIO) constraints.	Process
EQ_ComBal	Balance equation: if Primal is selected, value of LHS of that commodity's balance constraint (usually 0). If Dual is selected, shadow price of the balance constraint.	Commodity
EQ_PEAK	Peak equation (electricity). Same as EQ_ComBal, but for the Peak constraint.	Commodity
NRG_INT	Energy Intensity of a sector	Demand category
SYSCOST	Value of objective function.	None
VAR_ACT	Activity variable of a Process	Process
VAR_CAP	Capacity Variable of a Process	Process
VAR_MEDG	Increase in elastic demand	Demand
VAR_MEDL	Decrease in elastic demand	Demand
VAR_EM	Total Emission of some emittant	Emission
VAR_FIN	Input flow into a process, calculated as the product of an input coefficient of some commodity into a process by the activity of the process.	Process, Commodity
VAR_FINFEQ	The fossil equivalent of the input flow associated with renewable energy carriers.	Process, Commodity
VAR_FOUT	Output flow from a process: calculated as the product of an output coefficient of some commodity out of a process by the activity of the process.	Process, Commodity
VAR_INV	Investment variable of a process	Process

* In addition, the scenario, time period, time slice, vintage, solution value, and region are implicit dimensions for all attributes.

Appendix C. VEDA-BE versions

Name	Characteristics/Observations
Viewer	<ul style="list-style-type: none"> • Views only cube files provided from VEDA-Complete/Developer or KanORS. • No save of restructured cubes, no new tables, no export. • This will be similar to the www demo, but easily installable locally. • Used for SAGE cubes of inputs and results.
Lite	<ul style="list-style-type: none"> • A dedicated loaded VD application. • No import facility. • MDB provided by VEDA-Complete/Developer or KanORS. • Full features to create new tables, unit conversion, export etc. • Can be used w/ multiple applications. • Used for SAGE input and results.
Standard	<ul style="list-style-type: none"> • Only supports an existing structure (thus no new database option). • Distributed with initial MDB provided by Complete/Developer or KanORS. • Can import new sets of results. • Full features to create new tables, unit conversion, export etc. • This is what ETSAP/Outreach is entitled to.¹⁵
Complete	<ul style="list-style-type: none"> • Can construct new databases based on the VD file headers. • This is what is sold with GAMS.
Developer	<ul style="list-style-type: none"> • Can create the input for VEDA-Viewer, both for the local and www variety, as well as Lite/Standard users of their applications (though users need to purchase VEDA-Lite/Standard). • Hill Associates

¹⁵ A set of MARKAL/TIMES “build” tables will be provided for systems intended for use with these models.