Graphical Interface Manager 6 Reference manual

Although care has been taken in preparing the information contained in this site, all material that is provided here or could be reached by using the website as a starting point is supplied “as is” without warranty of quality or accuracy or of any other kind.
Neither Tizio B.V, Netherlands Environmental Assessment Agency, nor any author or supplier contributing to this manual is responsible for any errors or omissions in any information provided or the results obtained from the use of such information.

website: www.my-m.eu
e-mail: info@my-m.eu

Contents

Part 2 Designing Views 37

1. Introduction

This manual describes the various dialogs and functions available in the Graphical Interface Manager (GIM) of the MyM package. Please refer to the Getting Started [3] for a general overview.

This manual is not intended to be read from start to finish, as it consists of a collation of individual help documents that can be directly accessed from the GIM (by pressing F1 while pointing to an interface item). As these help documents are fairly independent of each other, they are rather redundant when put together.

Developer versions of MyM are available for Windows XP and higher. They require the Visual Studio 2005 compiler.
Runtime versions run on Windows without any auxiliary software. On-line documentation is provided by a built in HTML browser.

What is new in version 6

Version 6 is a completely new version of the GIM. Most things have remained the same. We will outline the differences in the next paragraphs

Menus

- The menubar has been reorganized.
- Menus that were previously only available through shortcuts have a place in the menubar.
- The magic context menu in the screen's border was removed, and is now available in the menubar.
- "Show arrows" was renamed to Dependencies.
- New menu: Check for Updates, which checks if there is a new version available of the software.

Interaction

- M_QUICKSHIFT is not supported anymore.
  It is now replaced by the doubleclick function of the mouse.
- Doubleclick corresponds to opening/closing a widget and Shift+Doubleclick zooms in and out.
- Design mode is now really a design mode, i.e. you can not edit a graph or slider when you are not in use mode.
- There is a visual indication when you resize an object.
- The context menu in a graph was simplified. The option drag graph/scale x,y was removed. The user can adjust these settings in the Domain and Range Dialog.
- When editing a graph, the coordinates x,y, are shown in the status bar instead of in the graph itself.
- The time is not shown in the header of a widget anymore, if the Time Manager is visible.

Dialogs

- All dialogs are shown and remain on top of the viewer.
- GIM has now its own HTML Browser built in.
- Dialogs do not close when the user switches from Design to Use mode. Only the advanced panel is hidden.
- In the Range Dialog, Qualifiers are shown in the advanced panel.
- All templates can be selected with a Combobox, thus the user can either type a name or select it from the list.
- The cancel button in the Qualifier panel has been deleted.
- In the Dimension Dialog, the color of a class is now also shown in the class selection list, giving the user a visual feedback on what is selectable.
- The Table editor has been improved, values can be edited directly in the cells.
- The option “Single Dialog” has been removed in the Viewer Options Dialog.

Views

- In previous MyM versions, views were split, but using the same view structure. In version 6, views are split by creating a copy of the view. Thus splitting views will result in more views in the View Manager.

Note to old users

You must recompile your old MyM models, otherwise models will fail to load in version 6. Also, gif pictures are not supported anymore, please convert them into jpeg or png.

Ways to start GIM

There are two ways to start GIM:

- Type MGIM on the command-line or use the File-Manager and double-click MGIM.EXE.
- Double click a MyM model (extension .mdl).
  The GIM is the default open action for compiled MyM models.

On the command-line various option flags can be used:

`mgim` [*modelname*] {[−*option* [*arg*]]}

- `-help`
  Prints the available options. Always do this first, options may have changed.
- `-view` *view_name*
  Loads the view *view_name*. It is possible to load more than one view by using this option multiple times.
- `-htm`
  Has been changed from the previous version, where it was used to select the browser. Now the argument behind -htm is ignored. The GIM has it’s own integrated HTML browser.
- `-noint`
  Turns the interpolation option off. The default value is “on.

- `-noaut`
  Prevents the simulator from running (which is the default) during startup.
- `-size wxh` changes the default size of the Model Viewer.
- `-scen` *scen_name*
  Loads the (previously) stored scenario called *scen_name*
- `-scdef` *scen_name*
  Initially the scenario that is in memory during startup is called Default. You can change this initial name by using this option.
- `-version`
  Prints the version of MyM

On-line documentation

MyM provides on-line help for both system and model documentation. Documentation must conform to the HTML standard for hypertext documentation. As this is the standard used on the Web, MyM documentation can be explored using any HTML browser (Internet Explorer, Firefox, Safari etc.).

Help is called up by placing the mouse on the item of interest and hitting the F1 key. If the item is a model component (i.e some item in the Viewer), the help viewer chosen will show documentation provided by the model developer for that item when the user presses F1. If the mouse is on a certain dialog item in GIM, the appropriate entry in this manual will be presented. Other manuals for MyM are also available on-line (MyM Language Reference manual [1], MyM Language Tutorial [2]).
This version of MyM provides its own Help browser.

Manual outline

The base of the graphical user interface is formed by the Viewer. In the viewer the designer can create an interface to the model. The model developer can debug the model visually. The end user can run the model with predefined user interfaces.

First we introduce GIM's Viewer, followed by a description of the components of the Designer, the other important component of GIM's viewer. With the Designer, the modeller can create multiple interfaces to any MyM model.

The last part of the manual states other useful information like GIM's on-line documentation systems, a more thorough introduction to views and their use, scenarios, and the macro language available for GIM.

Part 1 GIM Viewer

2. GIM Viewer

The Viewer provides a window for any number of views on a particular model. A view consists of a number of widgets for model variables. In the simplest view, all widgets in a view are displayed at the same level, providing a kind of dashboard of the model.

Widgets can be opened or closed. To inspect a graph more closely a widget can be blown up to full window-size by zooming in and input widgets can be manipulated to change the value of the corresponding model variable.

As models become larger, the need for more structure will arise. This structure can be added by organizing widgets into subsets. By repeating this process at different levels, a hierarchy of strict subsets (i.e. a tree) can be created. Views can thus be set up as modular block diagrams which can be explored by zooming in on specific submodules.

To allow multiple views to be displayed and manipulated simultaneously, the Viewer window can be split into any number of non-overlapping panels. Each of these panels provides a separate view on the model. There are three situations where this can come in handy:

a) As an end user you would like to see the same view with different scenario data in the 2 panels side by side.
b) As an end user you would like to see 2 different views (thus 2 different viewpoints) on the same model side by side,.
c) You would like to have the same view with the same data, but you would like to see different levels of detail in both.

The widgets displayed by the Viewer are controlled by direct manipulation through mouse actions. The menubar at the top of the MyM Viewer provides access to a number of tools for more general actions.

3. Menubar

GIM's Viewer consists of two parts:

- a menubar and
- the part where the actual interface is located.

By clicking the underlined letter together with the Alt key on your keyboard anywhere in the Viewer, the corresponding menu rolls down and you can select any menu item. E.g. Alt+F opens the File roll-down menu.

The rolldown menus are described in succession:

File

- **Open Model** - Loads another model in GIM's Viewer.
- **Print** - Prints the view(s) currently displayed in the Model Viewer in the active Screen.
- **Exit**

Mode

- **Design** — Changes the mode of the Viewer to designer. By selecting Design the more advanced options in the dialogs become avialable and the view can be edited by editing or adding widgets to the views displayed.
- **Use** — Switches the mode of the Viewer to Use. In use mode you can present your model, without having the possibility to change shapes, positions or linkages.
- **Freeze** — When a model value (e.g. a graph) is changed, the model will immediately be run and the new calculated values of the model in GIM's viewer. If you want to edit multiple variables in

the model without an update after each adjustment, you should press the Freeze button. The views do not show a consistent picture anymore. If you unmark the Freeze button the model will run and show the consistent state of the model for the edited values.

- **Break** — The break option breaks into the simulation and stops it.
  Most simulations in GIM run instantly when a user changes a value and give immediate feedback. GIM also offers you the possibility to show intermediate states while running the simulation and break the simulation.

New

- **Container** — Enables the mode to add a container widget to the graphical user interface.
- **Variable** — Opens the Variable Manager (See section “Adding Widgets” on page 40) where you can select the model variables to be added to the selected view.
- **Selector** — Enables the mode to add a selector widget to a view. A selector widget is linked to a model variable’s dimension. If it is linked it serves as a radio or push button button box for the classes in that dimension.

Edit

- **Layout** — Opens the "Layout Dialog" (see section "Layout Dialog" on page 46). This Dialog is only available in Design mode.
- **Domain** — Opens the "Domain Dialog" (see section "Domain Dialog" on page 50). The Domain Dialog consists of two parts. One part that is always visible and editable, and a more advanced panel that is only visible in Design mode.
- **Dimension** — Opens the "Dimension Dialog" (see section "Dimension Dialog" on page 52). In the Dimension Dialog other selections can be made and the attributes of the classes can be changed. This Dialog also has a basic and a more advanced panel that is only accessible in design mode.
- **Range** — Opens the "Range Dialog" (section "Range Dialog" on page 55). In the Range Dialog you can edit the range of the model variable and design mode qualifiers can be added or changed.
- **Widget** — Opens the "Widget Dialog" (see section "Widget Dialog" on page 44). In the Widget Dialog you can change the graphical presentation of the model variable or container.
- **Copy** — Selects the currently selected widget to be the prototype for copying. For more information about how to copy widget attributes consult section "Copying Attributes" on page 61.
- **Paste** — Pastes the selected attributes from the prototype widget to the selected widget.
- **Delete** — Deletes the selected widget. *Delete* only applies to a single widget.

Tools

- **Scenario Manager** (See section "Scenario Manager" on page 29) — The Scenario manager allows users to load and save scenario data sets.
- **Time Manager** — Replays a model run (See section "Time Manager" on page 34). The Time Manager displays the current time in de GIM Viewer.
- **Equation swatch** — Shows the equation for a variable as specified in the underlying simulation model. Not available for non-model variables or input variables.

- **Table Editor** — Opens the table editor for a model variable. If the variable is not an input variable the commit button in the table editor is disabled.

View

- **Load** — Loads a view in the current selected screen.
- **Save** — Shows the on-line reference manual
- **View Manager** - Load and save views from disk (See section “View Manager” on page 23).
- **New** — Creates a new empty view.
- **Default** — Creates a default view from all variables available and create a new view in the current selected screen.
- **Next** — Selects the next available view.
- **Previous** — Selects the previous available view.
- **Dependencies** — Checkable option that shows or hides model based variable dependencies by drawing arrows between the variables.

Options

- **Model** — Shows the model specific options.
- **Viewer** — Shows the viewer specific options.
- **Save** — Saves the options to file in *modelname*.mopl.

Help

- **Viewer** — Shows the on-line GIM reference manual.
- **Language** — Shows the on-line MyM Language Reference Manual.
- **About** — Shows the about notes.
- **Check for updates** — Checks if there is a newer version available.

4. Exploring and using a model

To explore and use a model, the model and at least one view for that model should be loaded. A View is graphical representation of the model, where the model variables can be organized in containers.

Run-time versions of MyM models will normally start up with model and view(s) already loaded. Use the Model and View Managers if this is not the case, if no view has been selected or if a new view is desired. To use the model, make sure that the GIM is in Use mode (check under the Mode menu).

Opening a model

Unless you are using a GIM in which the model has been statically linked (see Getting Started manual), you can switch between models in the GIM. Select the *Open Model...* menu item in the File roll-down menu in the menu bar. A file browser appears where you can select a MyM model. MyM models have the extension .mdl.

If a new MyM model is selected in the Open model dialog, all memory occupied by the previous model is cleared, including views, datasets etc.

Another way to open a model is to use the model as command-line argument during startup. You can link the GIM to a MyM model by providing the modelname as first agument.

Mouse Actions

The basic mouse actions are:

- **LEFT**
  Select and resize a Widget, edit a Graph
- **RIGHT**
  Pop-up menu
- **DOUBLE CLICK**
  Open / Close the Widget
- **Shift+DOUBLE CLICK**
  Zoom in / Zoom out
- **MIDDLE (Alt-Left under Windows)**
  Move a widget. Only available in *Design* mode.

The Pop-menu depends on the state of the widget (opened or closed for instance). It may contain the following options:

- **Close**
- **Open**
- **Open All** — Opens all widgets in the selected container.
- **Close All** — Closes all widgets in the container.
- **Reset** — Resets all data in container to their default values.
- **Load...** — Loads a file with MyM data, resetting all variables contained in this widget and for which new values are found in the file.
- **Save...** — Saves all variables in container to file in MyM data format.

When the mouse is on the data area of a selected widget, the menu can contain a number of options that change the behaviour of Mouse-LEFT:

- **Show value** — Prints the X and Y value at the mouse cursor.
- **Edit Free** — Edits individual values by dragging them.
- **Edit Line** — Edits values by dragging the first and last value and linearly interpolating any intermediate values.
- **Edit Expo** — Edits values by dragging first and last value and exponentially interpolating intermediate values.
- **Table edit** — Pops up the Table editor for a selected model variable.

These various edit modes are only available for input variables, i.e. variables that do not depend on other variables (and that have not been "frozen" by the designer of this view).

Only those points on an input graph can be edited for which a default value has been specified. So if the model builder has only specified default values for the start and end time points, the user can only change those two and all intermediate steps will be set by GIM using linear interpolation.

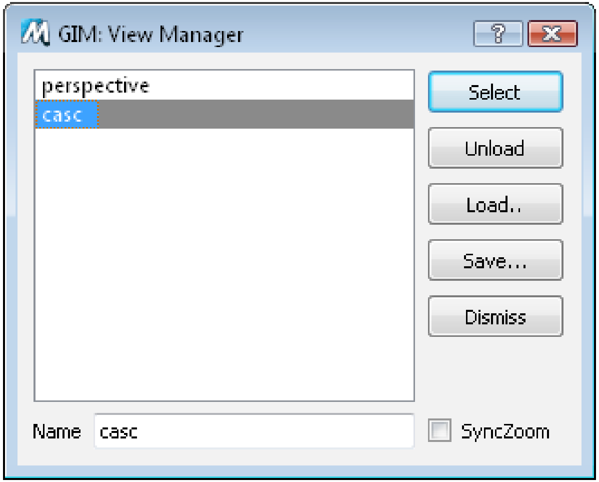
Keyboard action

Currently several short-cuts can be used to speed up interaction. The keys that are supported include:

- **Ctrl+A**— Pops up Domain Dialog.
- **Ctrl+B** — Interrupts simulator while running.
- **Ctrl+C** — Selects prototype for copying widget attributes.
- **Ctrl+D** — Switch to design mode.
- **Ctrl+E** — Shows the formula.
- **Ctrl+F** — Toggles the freeze mode on/off.
- **Ctrl+G** — Pops up Viewer Options Dialog.
- **Ctrl+I** — Pops up Dimension Dialog.
- **Ctrl+L** — Pops up Layout Dialog.
- **Ctrl+M** — Pops up the Model Manager Dialog.
- **Ctrl+N** — Shows the next view in the Model Viewer.
- **Ctrl+O** — Loads a view
- **Ctrl+P** — Displays the Print Setup Dialog.
- **Ctrl+R** — Pops up Range Dialog.
- **Ctrl+S** — Saves a view
- **Ctrl+T** — Pops up the Time Manager.
- **Ctrl+U** — Switch to use mode.
- **Ctrl+V** — Pastes selected attibutes from prototype widget to selected widget.
- **Ctrl+W** — Pops up Widget Dialog.
- **Ctrl+X** — Deletes selected widget
- **Ctrl+Y** — Pops up the View Manager. For more information on GIM's Viewer consult section "View Manager" on page 23.
- **SHIFT+Ctrl+P** — Opens the previous view in row (as listed in the view manager) for the selected panel.
- **Ctrl+TAB** — Opens the next view in line for the selected panel.
- **F1** — Shows help information. If the F1 key is pressed inside the Model Viewer, any information enterd by the designer of the view is displayed in the helpviewer. If the F1 key is pushed inside a GIM system dialog, help for that dialog and items is displayed in the helpviewer.

5. View Manager

The View Manager is used to load views from disk, to save views that have been changed to disk and to select or unload views that are already loaded in memory (Note: the view is deleted from memory, not from disk!). The View Manager is called up by selecting the View Manager item in the View roll-down menu in the menubar.



The selector shows all currently loaded views, using colored highlighting for the view selected in the active panel. The views that are currently visible on the screen are indicating by a dark gray background. Only views that have no highlighting can be selected.
The name of this view is also shown in the View name input box at the bottom of the View Manager. Views are stored on disk in a special MyM format. This file should not be altered by the user. A view definition file has extension .VDF and contains information about the widgets in a view, their attributes and the references and descriptions of the templates used.

Select

To select another view simply select the entry in the browser and click on Select. If you want to select a view for another panel, first click on a widget in that panel to update the list.

Unload

A view can be removed from memory by selecting it and clicking on Unload. At least one view has to remain in memory.

Load...

To load a view from disk, click on Load... to call up the File Selector. If a view is loaded for a model that has been changed, the descriptions for the widgets or templates might not be up to date. The View Manager tries to fix the problems that occur.

Save...

Views can be (re)stored on disk by selecting the view to be saved and clicking on Save to call up the File Browser.

Dismiss

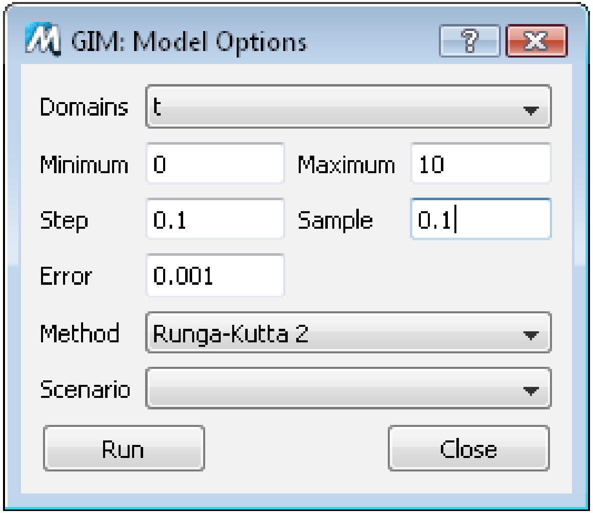
Closes the view manager by pressing Dismiss.

6. Printing a View

To make a hardcopy of an interface click on Print item in the File roll-down menu in the menubar.
Print sends the image on screen to the installed Printers.
Use zooming to select a subset of widgets if desired.
If you want to use the output for editing, you should send the output to a postscript printer and then select a file, or send it directly to a postscript printer, like Adobe's Acrobat or e.g. CutePDF writer.

7. Model Options

The model options dialog controls the settings of the simulator. Use the model manager to change the domain attributes of the variables in the model. The settings that are displayed in this dialog are from the domain variable chosen in the *Domains* choice at the top of the dialog.


GIM: Model Options

Domains: t
Minimum: 0
Maximum: 10
Step: 0.1
Sample: 0.1
Error: 0.001
Method: Runga-Kutta 2
Scenario:

Domains

The *Domains* choice displays one domain from the lists of domains that are used in the currently loaded model. Click on the choice with the left mouse button to select another domain variable.

Minimum

Defines the minimum of the simulation time. This minimum refers to all domains and can not be changed for one simulation domain only. If you want to define a new minimum just type the new value.

Maximum

Defines the maximum of the simulation time. This maximum defines the maximum value for all simulation domains in the model.

Step

The step size determines the step for which the currently selected domain in Domains is calculated. Type the new value to change the value.

Sample

Sample determines the size for which data of the currently selected domain in Domains is stored. Sample can be a multiple of Step. Change the value by typing a new value.

Method

Currently four integration methods are supported, i.e. Euler, RK2, RK4, and RK5, an adaptive step-size algorithm. For more information about the integration methods consult the Mym Language Reference manual.

Scenario

MyM has the ability to load and run multiple scenarios. With the Scenario choice you can select for which scenario you want to run the model. By default this choice is set to the scenario that corresponds with the selected panel.

Run

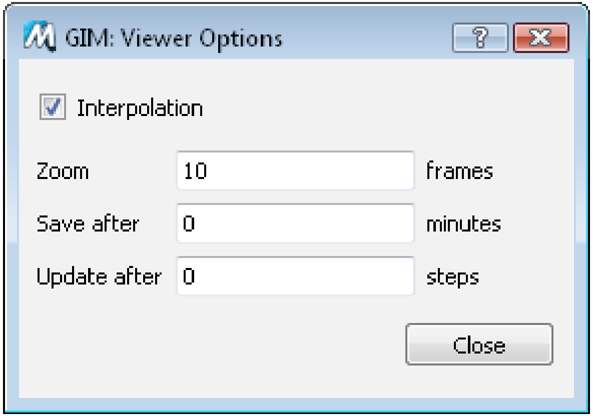
If you have changed some settings for any simulation domain you should press the Run button to simulate the model with its new settings.

Ok

If you click on Ok the Model Manager disappears.

8. Viewer Options

The Viewer Option dialog has toggles for the following options:



☑ Interpolation

Zoom 10 frames

Save after 0 minutes

Update after 0 steps

Interpolation

Interpolation determines if intermediate values should be interpolated in case they are not known. Otherwise discrete values are used.

Zoom

The Zoom factor determines the number of intermediate frames drawn while zooming. Set to zero, no intermediate frames are drawn.

Save after

GIM provides an auto save mechanism for views. Change the value for Save after by typing the new desired number of minutes between two saves. If the number of minutes is set to zero, auto saving is disabled.

Close

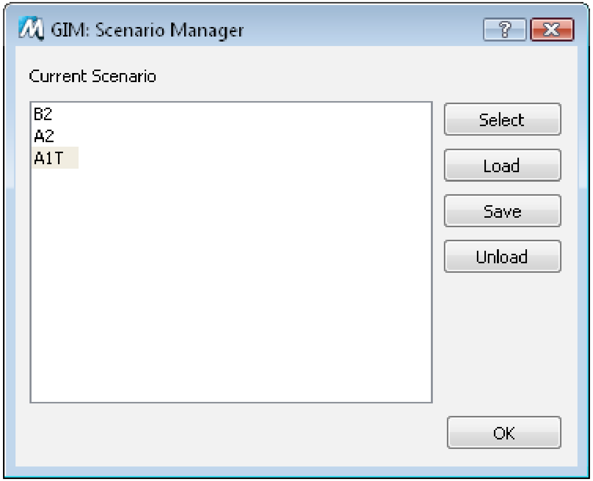
The option panel disappears by clicking on Close.

9. Scenario Manager

The scenario manager gives you the option to load multiple scenarios (see section “Scenarios” on page 64) in the GIM Viewer, unload scenarios, select and save scenarios. All scenarios stored on disk can be shown in the GIM Viewer by loading them into memory, with a maximum of 10 simultaneously.
At start up the Scenario Manager highlights the selected scenario for the currently selected panel in the GIM Viewer.
Graphs and Bar-charts have an additional scenario dimension. Use this to display multiple scenarios in one graph.

Load...

Opens the Load Scenario Dialog. All scenarios previously saved are listed in this dialog by their labels. Consult the section “Load Scenario Dialog” on page 31 for more information.



Select

Selects the selected scenario in the Scenario Manager to be displayed in the current selected panel in the GIM Viewer. All scenarios that are listed in the Scenario browser can be selected for display. If a scenario is not present use the Load... button to load the chosen scenario into memory.

Save...

Saves the currently selected scenario in the scenario Manager to disk. If you press the Save button a Save Scenario Dialog (See section “Save Scenario Dialog” on page 32) appears which you have to fill out. In this dialog you have to fill out the name, label, author, help document, color, and date. This information is stored and can be retrieved during a load scenario operation.

Unload

Unloads a loaded scenario from memory. All memory occupied by the selected scenario is freed and the scenario is removed from the list shown in the Scenario Manager. If the selected scenario is viewed in any panel in the GIM’s Viewer, these panels will show the next scenario in the scenario list.

OK

Closes the Scenario Manager.

10. Load Scenario Dialog

The Load Scenario Dialog gives you the ability to add scenarios to the list of loaded scenarios (see section "Scenarios" on page 64) in GIM's Scenario manager. Only the scenarios that are displayed in GIM's Scenario Manager can be viewed and compared in GIM's Viewer.

Available Scenarios

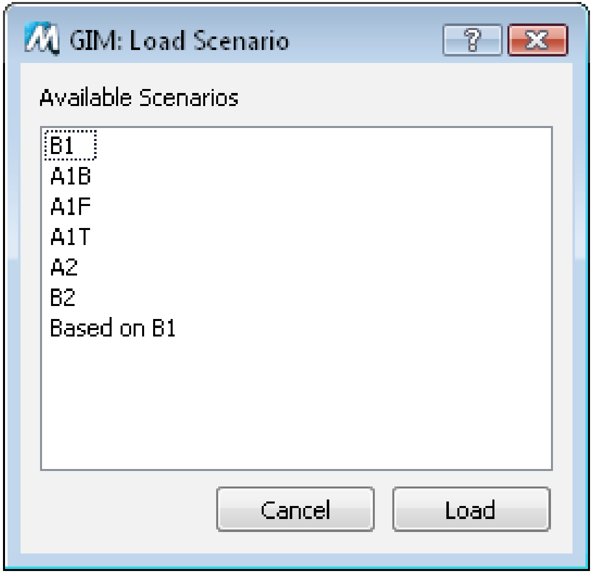
All scenarios that can be loaded are listed in this browser. The selected current scenario for loading is indicated by a high-lighted bar. By selecting any other entry in the browser you can load other scenarios.

Cancel

Closes the Load Scenario Dialog and does not load a scenario into memory.

Load

Loads the currently selected scenario into memory. All details of the scenario that were saved during Save Scenario are restored and used. For example, the color of the scenario is used to draw graphs with the corresponding legend.



GIM: Load Scenario

Available Scenarios

B1
A1B
A1F
A1T
A2
B2
Based on B1

11. Save Scenario Dialog

The Save scenario dialog appears if you press the Save... button in GIM's Scenario Manager. In the Save Scenario you have to fill out the details of the scenario (see section "Scenarios" on page 64), selected for saving. You have to fill out the following meta information:

Name

The name is a string that is passed on as the value for the $1 parameter in the scenario template. It can be used to index the saved scenarios. This name should be unique and appropriate).

Label

The label is used to show a more informative label than the name used for indexing. This label should typically be something like: "Baseline scenario Global CO_2 1%". NOTE: Spaces are allowed.

Author

To see who created a scenario.

Date

To enter the current date change the string in the Date input field.

Help

You can document your scenario more detailed in a help document. This document should confirm to the HTML-standard. Type the name of the scenario help document in the Help input field.

Color

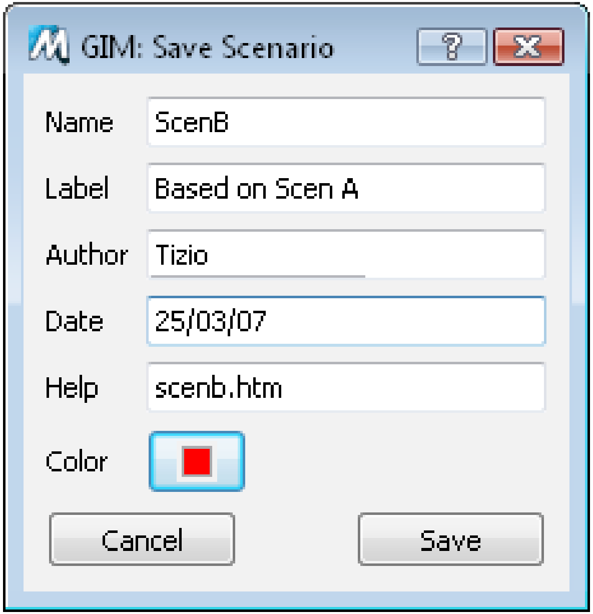
The color represents the color that is used to represent a scenario in a legend for a graph. You can change the default color by clicking on the the Color button. GIM's Colormap Browser appears where you can select another color.

Cancel

If you press the Cancel button, all changes are discarded and the Save Scenario dialog disappears.

Save

If you press the Confirm button, the current selected scenario is saved with the arguments filled out in the Save Scenario dialog.
After all information is saved you will return to GIM's Scenario Dialog.


GIM: Save Scenario

Name	ScenB
Label	Based on Scen A
Author	Tizio
Date	25/03/07
Help	scenb.htm
Color	🟥

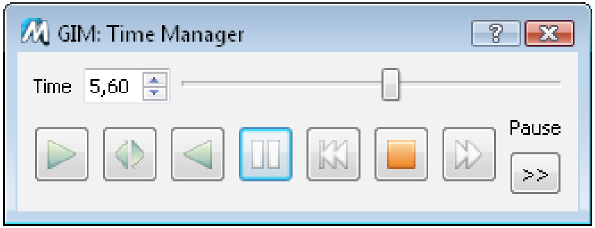
12. Time Manager

The Video or Time Manager is a DVD-player interface to step through the simulation time and animate the model results.

The Time Manager is built up from two panels. The first panel only displays the current time for which the data is displayed in the Model Viewer and the DVD-player like controls to step through the time axis.

The second panel of the time control contains additional sliders to control the start time, stop time, speed of animation and a toggle whether to cycle through time or not.

By pressing the more button the second panel with the more advanced controls are visible. At this point the more button turns into a less button, which if pressed hides the second panel.


GIM: Time Manager

Time 5,60

Current time

The time step currently displayed in the Model viewer. Click on *Next* or *Previous* to advance or play back to another frame at a time.

DVD-player controls

These buttons have a similar meaning to the buttons found on DVD players.

Reverse

Changes the play mode of the Time Manager. If the Time Manager is in play mode, the mode will change into Backward play and vice versa if the current mode is backward play.

Backward Play

If the Time Manager is in Backward play mode, the time will step to previous time frames. The step is determined by the speed factor and the minimal time step of the simulation (if speed=100).

Play

Starts playing if it is stopped or paused. The time is updated with a step that is determined by the minimal simulation time step in the model and the speed selected in Speed.

Pause

Toggle button that causes the Time Manager to pause playing. Press Pause again to resume playing.

Previous

Skips the current time to a previous time frame. The previous time frame is a multiple of the current speed factor and the minimal simulation step for the domain t.

Next

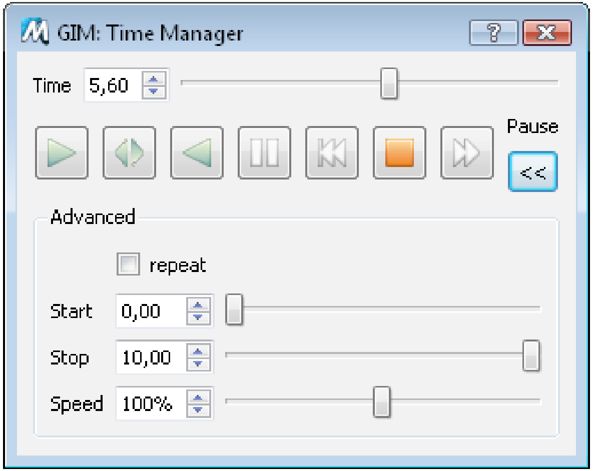
Navigates to the next time frame on the time-bar. The next time frame is determined by the step value which is a multiple of Speed and the minimal simulation time step.

Stop

Stops playing and resets time to the first time frame defined by Start.

Advanced Time Controls

The sliders and buttons that set the more advanced time manager settings are put in a separate panel. This panel appears if the '>>' button is pressed.



Time 5,60

□ repeat

Start 0,00

Stop 10,00

Speed 100%

Repeat

If the toggle *repeat* button is on, the Time Manager is put in repeat mode. Repeat causes the Time Manager to repeat continuously within the range of time, provided that the Pause toggle is not 'on'.

Start time

The time from which to start playing.

Stop time

The value at which animation should stop.

Speed

The speed can be varied between 0 and 200 percent. Higher numbers equal higher speeds. If the speed equals 100%, the step-size for sequential frames is the minimum time step defined in the model. To change the speed, use the slider or enter a specific value.

Part 2 Designing Views

13. Designing Views

GIM's Viewer can operate in two modes, i.e. *Design* and *Use* mode. The mode is changed by selecting the corresponding toggle in the *Mode* roll-down menu in the Viewer's menubar.

In *Use* mode the model is typically demonstrated or used by end-users that work with predefined interfaces (views).

In *Design* mode the views for the model are created, edited by designers or modellers. The creator can decide which variables are to be included in the interface. Note that interaction with the model can be achieved in both modes.

The *Design* mode is used to design Views for a MyM model: in *Design* mode you can add and delete items, edit their appearance and copy the appearance of one widget to other widgets.

If the viewer is started, it displays an empty view to the user. If the user wants to design a view, widgets can be added to the (empty) view. By selecting any of the items in the *New* roll-down menu (note that these items are only enabled if the viewer is in Design mode), and indicating the position and size in the Viewer (see section "Adding Widgets" on page 40) the interface is extended with the selected widget.

If *Design* mode is selected, the more button (>>) in the dialogs that are accesible through the Edit roll-down menu are enabled. By pressing this more button more advanced edit options become available, e.g. the possibility to link templates.

The following Dialogs can be used to edit a widget:

- Layout (see section "Layout Dialog" on page 46).
- Domain (see section "Domain Dialog" on page 50).
- Dimensions (see see section "Dimension Dialog" on page 52)
- Range (see section "Range Dialog" on page 55).

How to delete widgets is described in section "Deleting widgets" on page 62 and the copying mechanism for various attributes of a widget is described in section "Copying Attributes" on page 61.

14. Moving and scaling widgets

Objects are *moved* by using the middle mouse button. The middle Mouse button is simulated under Windows by using the combination ALT together with the left mouse button.
To move a widget press the middle mouse button inside the bounding box (not near one of the corners) and move the widget to its new position. A red outline indicates the widget at its new position. Widgets are not allowed to overlap. If the widget is moved and overlaps another widget, the red outline will not be drawn. If a Container widget is moved, all its contents are moved with it.

To *scale* a widget, pick one of its corners and drag it while keeping Mouse-Left depressed. The new size is indicated by an outline.

15. Adding Widgets

To add items to a view, select Add and select one of the available types by pressing the pop-up list next to the Add choice. The most common types are *Container* and *Variable*. Select the *Variable* type to add model variables to the view, represented as bar charts, line graphs or some other appropriate form (see section "Widget Types" on page 70). Use *Container* to draw boxes that can hold multiple variables. Containers are used to provide structure to what could otherwise become an unwieldy collection of individual graphs. By grouping variables in containers and combining these sets into higher-order containers, any hierarchically structured system diagram can be superimposed on a given model and will be directly reflected in the view being designed. Different views can be designed to reflect alternative organizations.

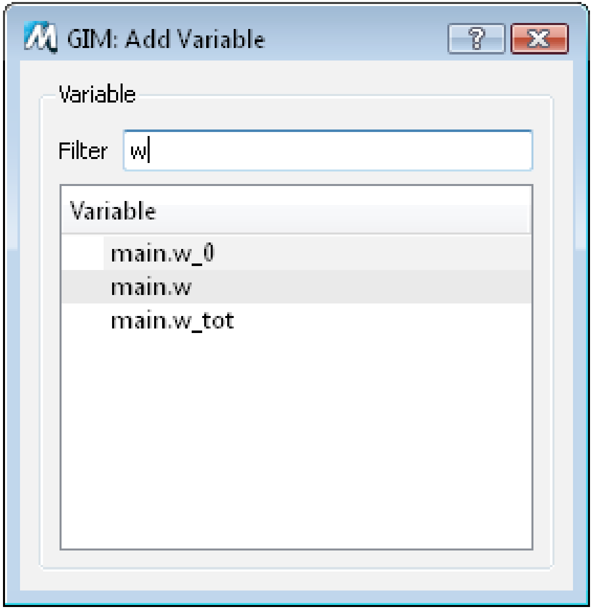
Adding a Container

A *container* is added by pressing Mouse-Left somewhere in the Viewer, dragging and releasing the mouse once the box is large enough. Any existing widgets that lie completely within its borders become part of the new container. Widgets can also be added later, moved between boxes by dragging or deleted. Multiple containers are added by keeping the SHIFT key pressed. To change the default appearance and title of containers, switch to Edit mode (see section "Editing Widget Attributes" on page 43).

Adding a Model Variable

To add model variables select *Variable* and use the Variable Browser to select a variable.

The Variable Browser lists the variables currently not in use in the selected view. To select a variable click on it, move the mouse to the Viewer and draw the outline by pressing Mouse-Left, dragging and releasing.
To remain in Add mode (i.e. to add more than one variable without having to select Add each time), press SHIFT while selecting a variable and drawing its outline.
To select the next variable, keep the SHIFT key pressed, and select another variable from the list. The selected variable is indicated by a colored background.



Filter w

Variable
main.w_0
main.w
main.w_tot

Filter

If the variable list is long, a substring of the variable name can be used to restrict the list to names containing that substring.

16. Adding a Selector

In some situations you might want to be able to stear the dimension selection without letting the user use the (rather complicated) Dimension Dialog.

In those situations you could include a so called '*Selector*' object to your view. These selectors can be linked to dimension templates as descibed in the "Scenario Template Files" on page 80.
A selector is represented by a button-box. By pushing one of the push buttons you can select or deselect a class in the corresponding dimension.

To add a dimension selector select *Selector* in the *New* drop-down menu. Then select a simension from the presented list and move the mouse to the Viewer and draw the outline by pressing Mouse-Left, dragging and releasing.

A dimension selector corresponds with one of the dimensions of variables in the model. Which dimension is determined by selecting the dimension template in the Dimension dialog (see "Dimension Dialog" on page 52).

More information about selectors can be found in the section "Selectors" on page 65.

17. Editing Widget Attributes

Widgets can be added to a view using the *New* roll-down menu. They all come up with a default appearance that can be changed.

All attributes of a widget are grouped into several templates. The different types of templates relate to the dialogs available. By using a specific dialog, e.g. a range dialog the user can change the appearance of the range.

How to use dialogs is explained in the next chapters.

18. Widget Dialog

To change the graph representation or displayed selection of dimensions, put the Viewer in Design mode and select a widget with Mouse-Left. A red outline indicates that the widget is selected. Choose *Edit|Widget* and the Widget Dialog will pop up. This panel contains the following lines:

Widget

If the widget represents a model variable, its model declaration is listed here.

Class

The widget is either a *container*, a *variable*, a *domain variable* (time dependent), or a *selector*.

Type

The type of graphical representation (bar chart, graph, map etc.). The options available under the selection menu depend on the class. The type is changed by Mouse-Left on the selection menu (see section “Widget Types” on page 70).

Subtype

Depending on the Type, different subtypes can be selected. For bar charts these include normal versus stacked, for sliders horizontal versus vertical and so on. To select an available subtype, click on the Subtype choice and pick one.

Dimensions

This browser can be used in two situations: to indicate which dimension to use on the x-axis when displaying variables with more than one dimension as a bar chart, and to indicate which dimensions to use on the two axes of a map when displaying variables with more than two dimensions.
Press the line in the browser for the dimension to be selected. For gridmaps two dimensions can be selected. To select another dimension of a model variable on your axes, firstly deselect a highlighted dimension and then selected the dimension want on one of the axes.

19. Attributes and Templates

The attributes of a widget is divided into several categories, i.e

- layout
- domain
- dimension
- range
- type

For these categories of attributes GIM supports *templates* to determine the presentation attributes of the widget.

Grouping these attributes in named templates has two advantages: it saves a lot of work when many widgets have to get the same appearance (which is almost always the case) and it allows the designer to create dynamic links between widgets.

By using the same dimension template for different variables with the same dimensions, the designer can ensure that whenever the user makes a particular selection on that dimension on one of these variables, all linked variables will also be restricted to the same selection.

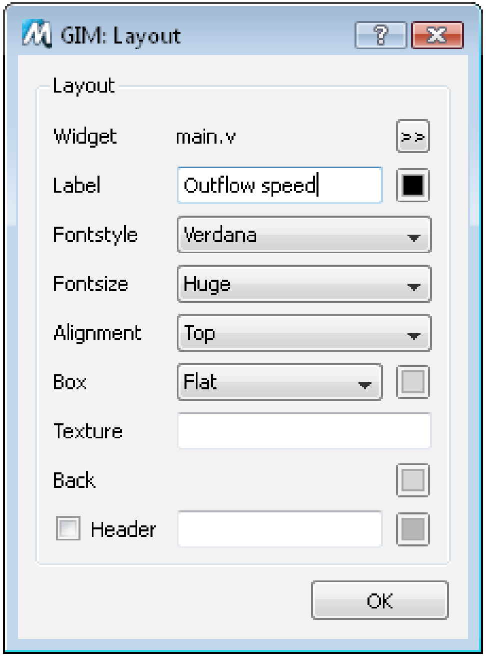
Similarly, when using the same range template, scaling or dragging one graph will automatically lead to similar updates of all other linked graphs.

The template editor can always be found in the advanced panel of the Dialogs. The template editor consists of an input field and a browser.

If you edit the name and this name already exists, then this other template will be used to determine the presentation of the widget, otherwise the name of the current template is changed.

If you select a template in the list of available templates (used by other widgets) then the current template is linked to the selected template.

20. Layout Dialog



Widget main.v

Label Outflow speed

Fontstyle Verdana

Fontsize Huge

Alignment Top

Box Flat

Texture

Back

□ Header

The layout dialog lists the attributes that can be changed for the current selected widget. The contents of the attributes will be updated when another widget is selected (by clicking the Left-Mouse button). Attributes include:

Label

For model variable widgets, the label is set initially to the model variable name. To change the label, type a string in the text field. Use the tilde (~) character for line breaks (i.e. labels that use multiple lines). Linefeeds only have effect when a widget is closed. If a widget is opened, a tilde will be interpreted as a space.
Click on the color button next to the label text field to change the color used for the label. This color is also used for the text along axes and for the legend (see section “Colormap” on page 58).

Font Style

The font selected is used for all text. The default font is Verdana. Some other fonts are also available.

Font Size

The maximum size to be used for text. The actual font size is determined by the space available for the text. The size ranges from small to huge.

Alignment

The position of the label (when open).

Box

The type of box used to contain the widget. Ten types are available varying in appearance up/down/flat, round/rectangle and shadowed or not.

Foreground Color

The *Front* button shows the color of the box. To change the foreground color, press it (see “Colormap” on page 58).

Texture

The background pattern for a widget. i.e. a file in PNG or JPEG format. To select a texture for the background of the header, enter the name of a PNG file. If the file can not be found, no texture is drawn and this input is discarded.

Background Color

The color of the background used for the data area. The button shows the current color. Clicking on it calls up the colormap browser.

Header

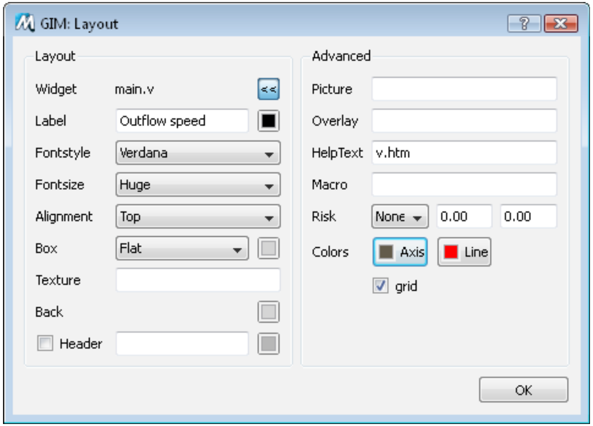
The header checkbox determines whether the top 10 percent of the widget should be reserved to display the name of the widget. A header can be colored independently from the rest of the widget.
The color of the header can be changed by clicking on the Header button.

Header Texture

To select a texture for the background of the header, enter the name of a picture. If the file can not be found no texture is drawn.

Advanced Layout Dialog

The advanced panel is shown when you click on the >> button. In the *advanced* part of the layout dialog you can edit:



GIM: Layout

Layout

Widget main.v

Label Outflow speed

Fontstyle Verdana

Fontsize Huge

Alignment Top

Box Flat

Texture

Back

□ Header

Advanced

Picture

Overlay

HelpText v.htm

Macro

Risk None 0.00 0.00

Colors Axis Line

☑ grid

Picture

To place a picture on a widget enter the name of a file containing a PNG or JPEG picture. If the filename is non-existing or you use the wrong format no action is taken.

Overlay

Vector Overlays are an alternative way to draw a picture on a widget. A vector overlay is a specification in terms of x,y coordinates of lines, polygons etc. The vector overlay is drawn on the background of a widget. To overlay a grid map of spatial data with the boundaries of countries, enter a VSF file describing these borders as polygons. For more information consult the section “Vector Specification Format” on page 82.

Help Text

The name of a HTML formatted file containing the Help text to be shown when a user hits F1 while the mouse is hovering over this widget. The path used is interpreted relative to *M_HELPROOT* (seesection “Environment Variables” on page 76).

Macro

In GIM Help texts, macros can be used to provide direct control over certain actions in the Viewer, such as zooming to a particular widget by clicking on a word or icon in the help text. These macros can also be linked to actions in the Viewer itself. So, a click on a button in the Viewer can be linked to a zooming action on a widget that is also part of that view. The same syntax is used as in help-macros. See section “Macro Language” on page 66 for more details.

Risk Area

Risk areas can be shown with differently colored regions in the background of a graph or chart. Choose None, X or Y-axis. The number of intervals depends on the number of ticks on the axes. Colors used run from blue (low) to red (high). Select the point where green starts by entering a value for the x axis (first number) or the y-axis (second number).

Axis and Line color

The color of the axes, grid and of the line used in zero-dimensional graphs (see section “Colormap” on page 58).

Input

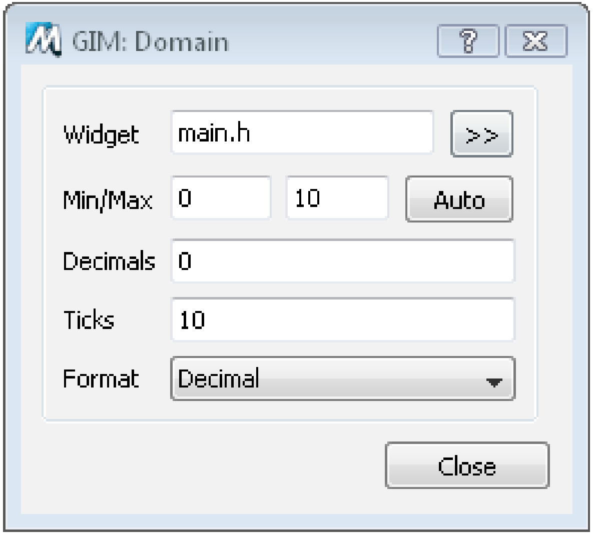
This option is only present if the widget represents an input variable. The toggle determines whether the user will actually be allowed to change its value (default is ‘on’).

Grid

Toggle to determine whether a grid will be shown in graphs or charts.

21. Domain Dialog

A domain is a template to specify the information needed to draw the x-axis. This domain can be adjusted in the domain dialog.



GIM: Domain

Widget main.h

Min/Max 0 10

Decimals 0

Ticks 10

Format Decimal

Minimum/Maximum

The Minimum and Maximum determine which interval within the actual domain is shown.

Auto

If you click on *Auto*, *Minimum* and *Maximum* values will be reset to the range in the model values.

Decimals

The number of decimals to be used for domain values.

Ticks

The maximum number of tickmarks to be drawn along the axis. The actual number used will depend on the available space.

Format

Currently, two formats are supported, *Decimal* and *Exponential*.

Advanced Domain Dialog

In the advanced panel (only available in design mode, and accesible through the >> button) you can edit:

Template

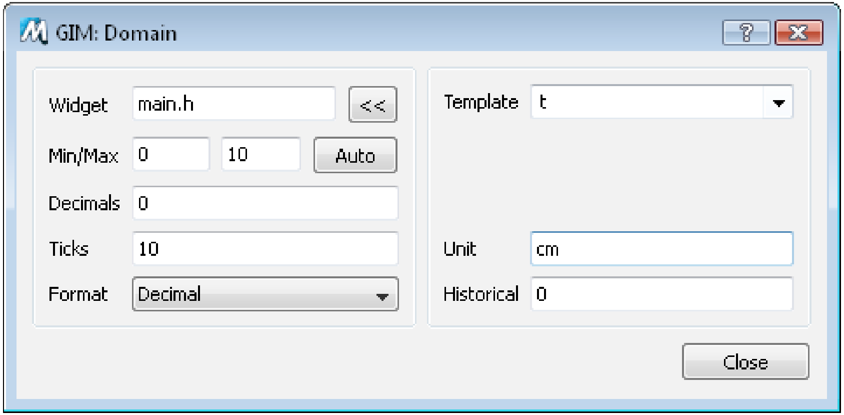
Indicates the currently used domain template. You can select another by typing the name of another template or select one of the listed domain templates in the browser.
If a new name is entered, the template is copied to a new instance with the given name.

Unit

The label to use on the domain axis.

Historical

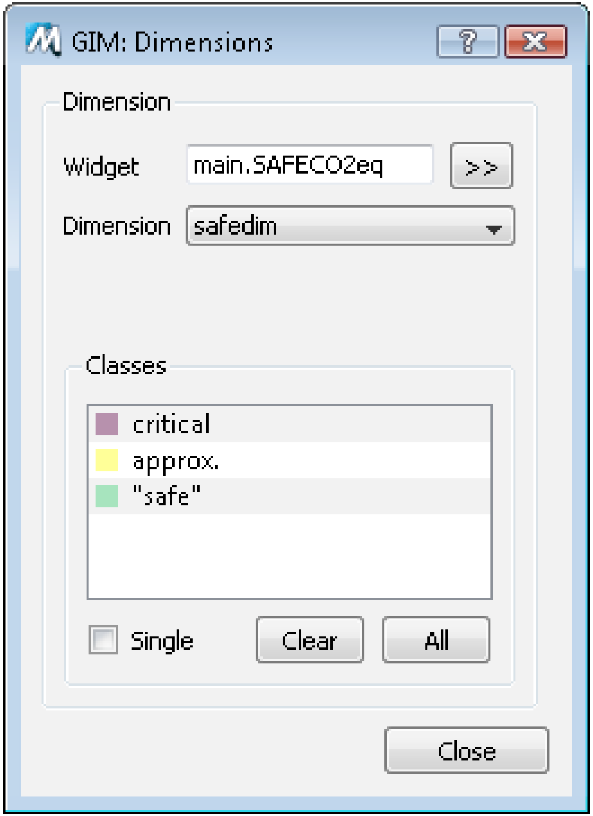
Prevents users from changing function values for any point before this historical value. E.g. if you set Historical to 1995, values before 1995 are historical and can not be edited.



Widget main.h
Min/Max 0 10
Decimals 0
Ticks 10
Format Decimal
Template t
Unit cm
Historical 0

22. Dimension Dialog

A dimension in GIM is a template which describes which elements and (graphical) attributes a dimension of a variable has. E.g. a variable x[2,3], has 2 dimensions. The first dimension has 2 classes, and the second dimension has 3 classes. To a variable in a view (e.g. a grid map widget) 2 dimension templates are linked for which the user can edit the colors, labels etc.
The top of the Dimensions dialog shows the name of the dimension template and where the variable in the view is linked to. Select the dimension to edit.


GIM: Dimensions

Dimension

Widget: main.SAFECO2eq

Dimension: safedim

Classes

- critical
- approx.
- "safe"

☐ Single

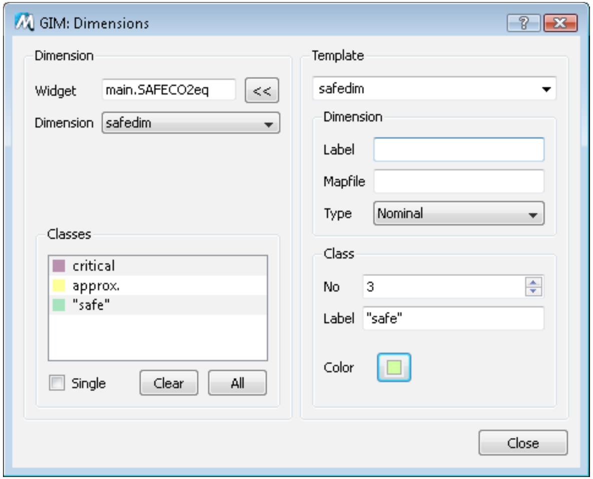
Dimension

To select a class click on an entry in the browser. An entry is deselected by clicking on it once more. If the browser is in *Single* mode (*Single* toggle is 'on'), only one class can be selected, if the *Single* toggle is off, any number can be selected. The *All* button selects all classes, the *Clear* button deselects all classes.

Edit Dimension Template

Template

Indicates the currently used dimension template for the selected dimension. Other dimension templates can be used by typing their names or selecting a template from the list of dimension templates that are compatible (have the same number of classes) displayed in the picklist. If a name is entered that is not available in the list, a new instance is created, and the dimension is unlinked.



Widget: main.SAFECO2eq
Dimension: safedim

Classes:
- 🟪 critical
- 🟨 approx.
- 🟩 "safe"

□ Single

Template: safedim

Label:
Mapfile:
Type: Nominal

No: 3
Label: "safe"

Label

The label is used for the legend or printed along the x-axis of bar charts or grid maps. The length of the label is used to determine the space available for the classes in a legend. Add extra spaces to fit longer class names in the legend.

Mapfile

The file name of a Vector Specification File (VSF) or a Raster Specification File (RSF). This file contains the shapes and positions of the cells to be used for individual values in Maps. For more information about the use of RSF and VSF files see section “Vector Specification Format” on page 82 or section “Raster Specification File” on page 83.

Type

For *ordinal* dimensions, classes are indicated by numbers, ranging from 1 to the size of the dimension. For *nominal* dimensions classes are indicated by the labels specified in the Class input field.

Class No

The *class no* determines the entry in the classes list that will be edited. You can edit the label and the color.

Class Label

By default, dimension values are indicated by numbers. To add a more informative *label*, like “Japan”, change the label in the text field. This label will be used only when you set the type of the dimension to Nominal. Pressing > selects the next class, pressing < selects the previous class (NOTE: Only the classes selected are available for editing here, to edit all of them, first press the All button in *Select* mode).

Class Color

To change the color code of a class, press the color button (see). The coloring of graphs and charts depends on the selected dimension values. For each value you can specify a color and this color is used to indicate that value in the graph. When a graph or chart represent a multi-dimensional class the colors of the various values are combined in a striped pattern. For example, a two-dimensional class with value RED on dimension one and value BLUE on dimension 2 will be drawn as a checkered blue/red line.

For bar charts the colors specified for the values of the dimension drawn along the x-axis are normally not used to stripe the charts, unless this dimension is the only dimension of that variable.

23. Range Dialog

The attributes that determine the appearance of the range of a widget can be shared among several widgets. In the range template you can edit the entries how the ranges are presented on the screen.

Unit

The label to be drawn along the y-axis. Its font is determined in the layout dialog. You can control the format that is used along the axis by dividing or multiplying by a number. E.g. if the numbers along the axis range from 0 to 5000, you could change it into 0 to 5 by adding *1000 in the unit field.

Min/Max

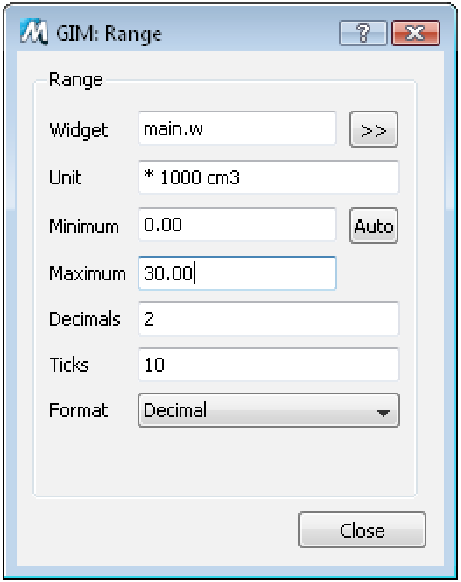
The minimum and maximum values to be shown.

Auto

Click on Auto to let GIM determine the minimum and maximum value for all variables using this range template.

Decimals

The number of *decimals* to be used.



Widget main.w

Unit * 1000 cm3

Minimum 0.00

Maximum 30.00

Decimals 2

Ticks 10

Format Decimal

Ticks

The maximum number of *tick marks* on the range-axis. Actual number depends on available space.

Format

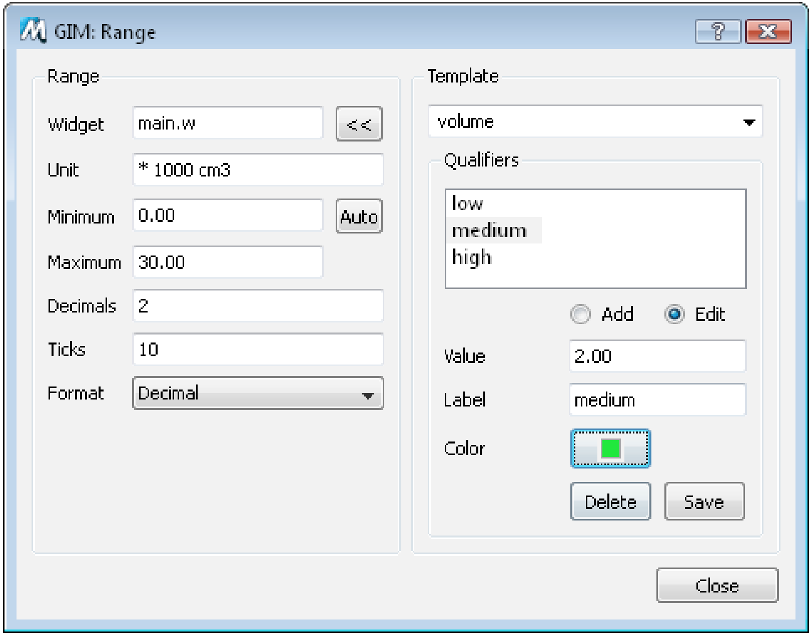
The format to print numeric values, i.e. *Decimal* or *Exponential*.

Advanced Range Dialog

In design mode the advanced panel of the range panel is also available and looks like this:

Template

Indicates the currently used *range template* for the selected widget. You can either type a name or select a range from the list. If the name is not in the list of range templates already present in the Viewer, all attributes are copied to a new range template with the new name.



Range

Widget: main.w

Unit: * 1000 cm3

Minimum: 0.00

Maximum: 30.00

Decimals: 2

Ticks: 10

Format: Decimal

Template: volume

Qualifiers:

- low
- medium
- high

○ Add ● Edit

Value: 2.00

Label: medium

Color: 🟩

Qualifiers

Qualifiers are named sub ranges. They can be given a color and used to indicate risk-areas in a graph or value ranges in maps. They can also be used to provide users with some qualitative support when setting values. The Qualifier selector lists the qualifiers in increasing order.

Add

Adds a new qualifier. The Value field is used to specify the upper bound of the sub range, its label and its color (see section “Colormap” on page 58). After a qualifier is added by pressing Save, the qualifier mode return to *Edit*.

Edit

If the *Edit* radio button is ‘on’, you can change the settings for the currently selected qualifier. The Qualifier panel appears, where you can change the following settings:

Value

The value to be used for the qualifier.

Label

The label (e.g. high) to be used for the qualifier.

Color

Defines the color that is used for the qualifier. In the situation where qualifiers are used to indicate intervals, the interval from the previous qualifier to the current qualifier is given this color. Click on the Color button to select another color (see section “Colormap” on page 58).

Delete

Removes the currently selected qualifier from the list of qualifiers.

Save

Saves the currently changed information of the selected qualifier from the list of qualifiers.

Close

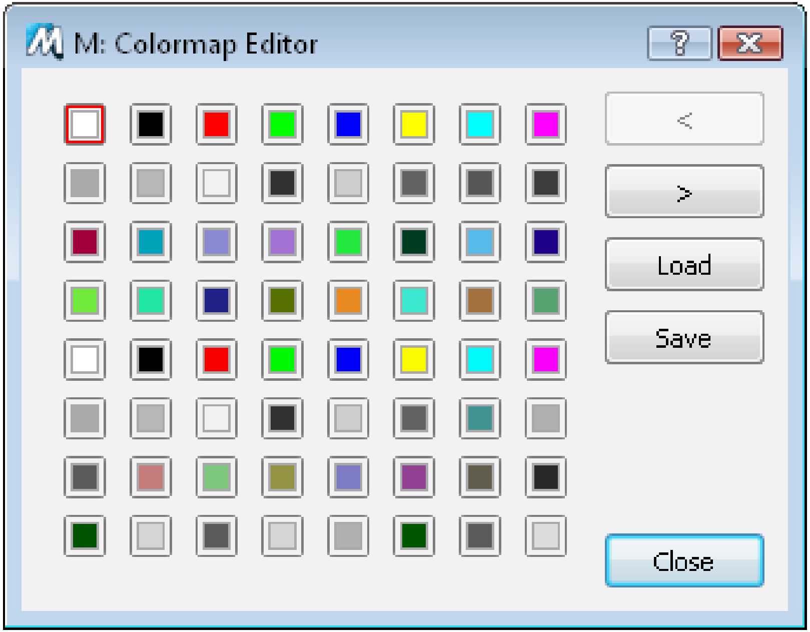
Changes are applied and the qualifiers are updated. If a qualifier is added it will be put in the qualifier browser, ordered by value.

24. Colormap

The colormap consists of 256 entries. The colormap editor is used to edit and select entries in this colormap. The Colormap Browser displays entries in blocks of 64 colors, so the colormap is divided into 4 blocks of 64 colors each.

< and >

The next and previous buttons at the right of the colormap grid are used to browse through the four pages of the colormap.



Color buttons

Click on a color button to select that color. Doubleclick on a colored button to change that color using the standard color change dialog.

Save

Saves the colormap in a file *model.*mcm. This colormap will be loaded whenever this *model* is loaded..

Load

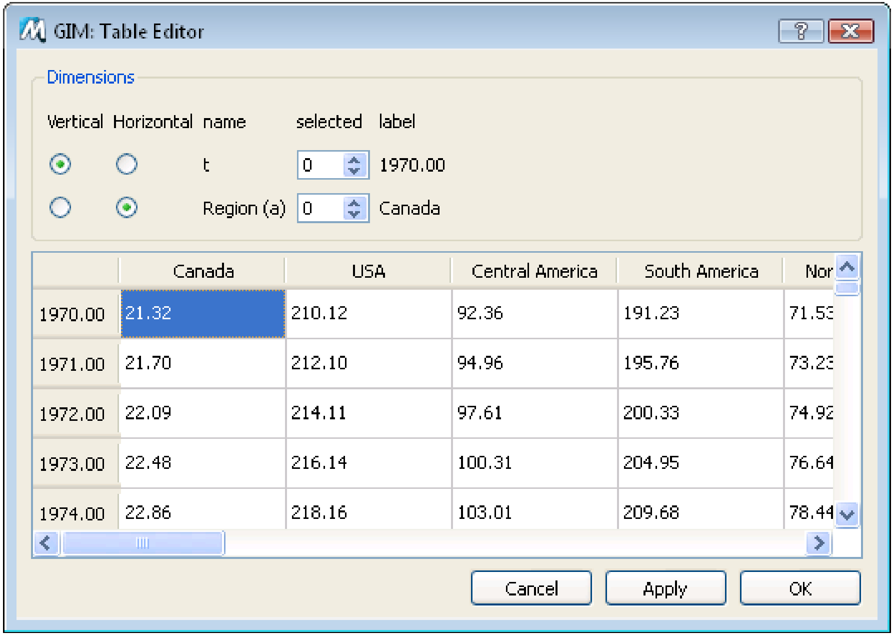
The map stored in the *modelname*.mcm file is restored.

Close

If you press the *Close* button the colormap browser disappears and the highlighted color is selected.

25. Table Editor

With GIM’s Table editor you have the option to set the values numerically. If the variable is multi-dimensional and/or time dependent, you have to specify which dimensions to display in the table.



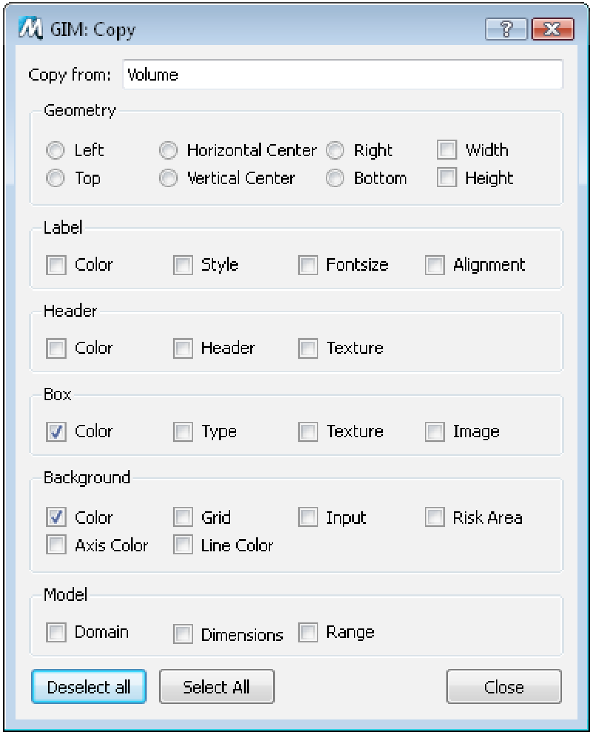
Vertical	Horizontal	name	selected	label
●	○	t	0	1970.00
○	●	Region (a)	0	Canada

	Canada	USA	Central America	South America	Nor
1970.00	21.32	210.12	92.36	191.23	71.53
1971.00	21.70	212.10	94.96	195.76	73.23
1972.00	22.09	214.11	97.61	200.33	74.92
1973.00	22.48	216.14	100.31	204.95	76.64
1974.00	22.86	218.16	103.01	209.68	78.44

26. Copying Attributes

To copy the attributes from a widget to other widgets, select the Copy menu item in Design mode. In the Copy Dialog each attribute that can be copied is represented by a toggle or radio button. The attributes are split into three sets:

- *Geometry*
  These attributes refer to the alignment and the width and height of a widget. Only one horizontal and one vertical type of alignment can be selected for a copy. Alignments or changes in size that cause overlaps or boundary crossings are ignored without warning.
- *Layout*
  The attributes that are also found in the layout panel in Edit mode. For these attributes see section “Layout Dialog” on page 46.
- *Templates*
  Template buttons under *Model* indicate whether a widget should use the same template for its domain, dimensions or range as the source widget. Copying dimension templates fails if the size and number of dimensions do not match.



Copy from: Volume

Geometry
○ Left ○ Horizontal Center ○ Right □ Width
○ Top ○ Vertical Center ○ Bottom □ Height

Label
□ Color □ Style □ Fontsize □ Alignment

Header
□ Color □ Header □ Texture

Box
☑ Color □ Type □ Texture □ Image

Background
☑ Color □ Grid □ Input □ Risk Area
□ Axis Color □ Line Color

Model
□ Domain □ Dimensions □ Range

Copying mechanism

The copying mechanism works in a Windows-style fashion. Copying works as follows: select a widget that serves as prototype to copy asttributes from. Once selected (indicated by a red frame) select *Copy* from the *Edit* menu. The copy panel appears where you can indicate which attributes should be copied to other widgets by turning on the corresponding toggle.

By selecting another widget and selecting Paste from the Edit menu, the selected attributes are copied to the last selected widget.

27. Deleting widgets

All widgets that are visible can be deleted from the view. A widget is deleted by selecting it in the view and choosing the *Cut* option in the *Edit* menu. Note that it is not possible to paste a widget; only attributes can be copied.

Every widget that has been deleted appears in the Variable list that pops-up if you choose *Variable* in the New pop-up menu. Because the widget has been deleted from the list all attributes are lost and the widget reappears if you add it again to the view with default settings.

If you want to delete all widgets inside a container, delete that container as described above, and you will be prompted to confirm your action (*Yes*). It is also possible to delete the selected box only (Select *No*).

Part 3 Advanced GIM Use

28. Scenarios

MyM includes the possibility to load, save and compare the results of different model runs in the graphical user interface. By using scenarios users can set all (or several) variables including output variables.

The Graphical user interface supports 10 scenarios to be loaded into memory. These scenarios can be displayed side-by-side or in one for linegraphs and bar-charts. By default the data that is defined in the model is the first scenario in memory.

You can reload scenarios from disk, delete them from memory or save them to disk by using the *Scenario Manager* (see “Scenario Manager” on page 29). If you save a scenario, all data specified by the scenario template file is saved to disk and an entry is the modelname.msd file is added to organize the saved scenarios. If you want to load a scenario this modelname.msd file is consulted to see which scenarios are available (previously stored).

Scenarios (or better: datasets) are loaded by using a scenario template. This template behaves as a filter. Only the variables mentioned are stored or retreived. The format of this file is described in the section “Scenario Template Files” on page 80.

Important: To use scenarios in the graphical user interface, the scenario template file should exist and have the same name as the model with the extension “.sce” and be put in the same directory as the model.

29. Selectors

It is possible to add button widgets to a view to represent a *Dimension selector*.

By adding a widget of class '*Selector*' you can have the same functionality as in the Dimension Dialog without having to open the dialog. The selector is represented by a button-box.
In order to link a selector to a specific dimension template you should enter the name of the dimension template in the Label field of the Layout panel for the given selector.

If you specify the dimension template to be in 'single' mode, only a single selection is allowed and the button box will act like a set of radio buttons.

30. Macro Language

GIM supports the use of Macros. Note that you have to run in ’*Use*’ mode to be able to run the macros.

Supported Macros

Currently the following macros are available:

- **gim** - causes actions in the GIM
- **play** - currently deprecated
- **shell** - causes a shell command to be run

All macros have the following format:

`{`*macro*`(arg1=value1[,argn=valuen]);},`

where macro = gim, shell. The number of arguments and their use depend heavily on the chosen macro. For more information about macros and their arguments consult the next subsections.

shell macro

Supported argument is

action

`shell(action=`*command*`)`

Causes the command to be run. E.g. on Unix: `shell(action=ls -al)` would list the files in the current directory.

gim macro

Supported arguments for *command* are:

- **action**
- **file**
- **model**
- **object**
- **scenario**
- **view**

The action argument determines which action in the GIM has to take place.

You can use one of the following values for action in combination with the listed arguments.

load

`gim(action=load,scenario=`*name*`)`

Loads a scenario.

`gim(action=load,view=`*name*`)`

Loads a view.

`gim(action=load,object=`*name*`,file=`*fname)*

Loads data from file *fname* into object *name*

`gim(action=load,model=`*name*`)`

Loads a new model into the GIM.

save

`gim(action=save,scenario=`*name*`)`

Saves a scenario.

`gim(action=save,object=`*name*`,file=`*fname*`)`

Saves data from object *name* into file *fname*.

zoomin

`gim(action=zoomin,object=`*name*`)`

Zooms in on object *name.* The name needs to be the internal name and not the label that is given to that object.

zoomout

`gim(action=zoomout,object=`*name*`)`

Zooms out on object *name.*

open

`gim(action=open,view=`*view_name*`,object=`*name*`)`

Opens a box for object *name* in the view *view_name*.

close

`gim(action=close,view=`*view_name*`,object=`*name*`)`

Closes box for object *name* in view *view_name*.

next

`gim(action=next[,view=`*view_name*`])`

If *view_name* is specified this view is opened in the root of the panels. If the view is not already loaded, the view is loaded from disk. If no view name is given, the root panel is replaced by the next in the list of views for that panel.

openall

`gim(action=openall[,view=`*vname*`[,object=`*oname*`])`

Opens all objects recursively. If no objectname is specified the root of the view is considered as root to open recursively, otherwise the object is searched which matches the *oname* specified. If no object matches this name no action is taken.

If no *vname* is specified, the openall applies to all views on screen, otherwise only for the view specified (if found!).

closeall

```
gim(action=closeall[,view=vname[,object=oname])
```

The same notes as for openall.

pause

```
gim(action=pause)
```

Normally for every macro you use in the GIM, the screen is redrawn. If you use multiple macros after another this might cause unneccesary updates. E.g. first you load a view (redraw), and then you want to zoom in on an object (redraw). To prevent unnecessary redraws you can use the pause action at the beginning of a multiple macro statement. The redraw is resumed after a `gim(action=resume)` statement.

E.g.

```
gim(action=pause);
...others_actions;
gim(action=resume)
```

print

```
gim(action=print)
```

Prints the views currently displayed in GIM's Viewer.

resume

```
gim(action=resume)
```

Causes a run event followed by a redraw event and causes all following macros to invoke a redraw event. For more information consult the notes of pause.

run

```
gim(action=run)
```

Runs the model.

set

```
gim(action=set,video={tnow|tmax|tmin|
                                  speed|repeat},value=x)
```

Sets the values in the video player. For repeat 0 indicates off and 1 on.

```
gim(action=set,video={play|bplay|pause|resume}
```

Sets the playing state of the video player.

How to specify macros in the GIM

Macros can be linked to widgets in the GIM. To specify a macro command just Select *Use->Design,* the widget where the macro actions should act on, and open the Layout Dialog (*Edit->Layout*).

In the advanced part of the Layout Dialog (use the >> button), the macro command can be added in the macro field. The macros are specified *without* the "hlpmacro:" statement descibed before.

Note: Macro commands are only active in *Use* mode.

31. Widget Types

Currently the following widgets are available

- *Container* to group other widgets.
- *Graphs* for domain dependent variables.
- *Buttons* for boolean variables and dimension selectors.
- *Dials, Sliders* and *Value Sliders* for scalar variables.
- *Charts(Bar, Pie)* for dimensional variables.
- *Grid maps* for multi-dimensional variables.
- *Maps* for spatial data (represented in MyM as one-dimensional variables).

Container

Containers are used to group other widgets (including other containers). This adds structure and hierarchy and allows users to apply the general interactions of zooming and opening/closing to groups as a whole. Containers have no subtypes, their appearance can only be changed by editing their Box-type. They allow the general interactions available for all widgets: *Zoom In*, *Zoom Out*, *Open* and *Close*; they also allow *Open All* and *Close All* to open or close all widgets contained by them.

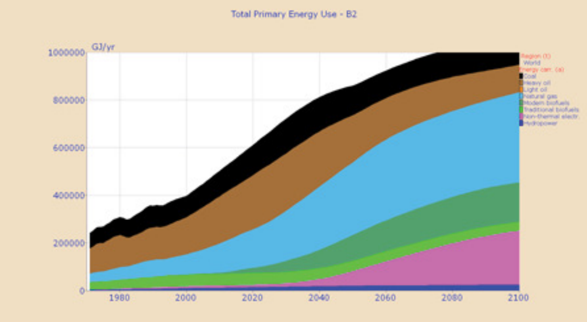
Graphs

A *graph* is a representation for domain-dependent variables.

The color of zero-dimensional graphs is determined by the color that is set in the object layout panel.

Multi-dimensional variables are represented by several lines, one for each class in the cartesian product of the selected dimension values.

When classes are based on more than one dimension, they are represented by multi-color lines composed of the colors of the contributing dimension values.



Subtypes

The subtypes include:

- *Line Graphs* where consecutive points are connected by a line.
- *Filled Graphs* when at most one dimension of a variable has multiple values. Values for different classes are stacked and indicated by filling the whole area between the value for a class and its previous class.
- *Dotted Line Graphs* where the sample points are indicated.
- *Dotted Graphs* in which only the sample points are plotted, without connecting lines.

Interaction

Press the right mouse button in the graph area to get a menu to select the mode for Mouse-Left or to perform a standard widget operation.

The following six modes are available for Mouse-Left:

- *Show value* — Shows the x and y values at the position of the mouse.
- *Edit Free* — Edits the graph by pulling the values of the graph to new positions.
- *Edit Line* — Edits the graph by drawing a straight line between the point where you start editing and the point where you stop editing.
- *Edit expo* — Edits the graph by drawing an exponential curve, displaying the exponent.

The default mode is *Show-value*.

Button

A widget that represents a boolean value. Pushing it with the mouse changes its value:

- *Push Button* — The button is ‘on’ (i.e. True) when the button is down.
- *Light Button* — A button with a small “light” inside. Pushing the button switches the light on. The color for “on” corresponds with col2 in the layout of the button, while the color for off corresponds with col1.

Dial

A widget that represents a scalar value. The dial can also be used to animate a time-dependent scalar value.

Slider

The *slider* widget shows a scalar value relative to some fixed bounds. When sliders are used for domain dependent variables, the slider represents the value of the variable at the time step shown in the Time Control (see section “Time Manager” on page 34).

Sub Types

The following types are available:

- *Vertical Slider*
  A vertical slider with a standard slide.
- *Horizontal slider*
  A horizontal slider with a standard slide.
- *Vertical filled Slider*
  A vertical slider with a bar slide
- *Horizontal Filled Slider*
  A horizontal slider with a bar slide.
- *Vertical Special Slider*
  A vertical slider with a fancy slide
- *Horizontal Special Slider*
  A horizontal slider with a fancy slide

Interaction

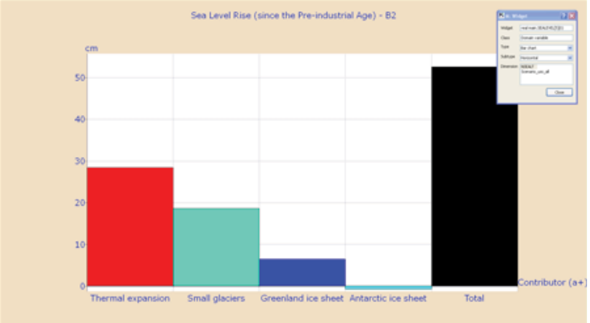
You can change the value of the slider by shifting the slide to a new position using the left mouse button. If you release the mouse button the new value is sent to the simulator.

Value Slider

A *value slider* is a slider with the value of the slider also indicated numerically.

Bar chart

Bar charts show the value (at time t) for selected dimension values.
The classes of one dimension are plotted along the x-axis. For multi-dimensional data, multiple bars are drawn for each of the primary classes. These bars can be stacked or filled with a pattern, where the colors of the stripes correspond with the combination of colors of the classes.



Sub types

The following types are available:

- *Bar chart*
- *Stacked Bar chart*

Interaction

Values for the selected classes can be changed by clicking the left mouse button in the chart area and dragging the charts to the desired values. When the mouse button is released, the new values are sent to the simulator and the simulation is rerun, unless it has been stopped.

Pie charts

Pie charts show the values for a dimension in the form of pies. For all other dimension the user should select only one class.

Grid

A grid can be used to represent multi-dimensional data in a two-dimensional matrix.

Subtypes

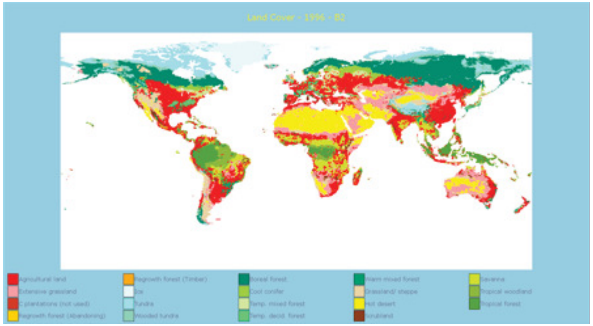
The values of the data cells are mapped to a color taken either from a “rainbow” scale or from the qualifier colors defined in the Range template for this widget.

Interaction

Data can not be edited.

Maps

Maps are used for spatial data that is represented in MyM by one-dimensional variables. Data values are indicated by a color taken either from a standard “rainbow” scale or from the colors specified in the range template (if the Qualifier subtype is chosen in the Widget Dialog). Data points are either depicted as a uniform rectangle (a Raster Map) or as a polygon (a Vector Map). While spatial data can also be stored in two-dimensional variables and presented as a Grid, this one-dimensional format is useful for sparse spatial data, as in the map shown, where only land-cells are computed. Storing these in a 720 * 360 matrix would put great demands on memory and display time.



Raster Map

Each value is depicted by a colored rectangle. The size of this rectangle and its position for each dimension value are specified in the Raster Specification Format file (RSF file, see section “Raster Specification File” on page 83

If you click the Right-mouse button in the raster map a menu appears where the specified regions in the raster file are listed in menu items. By selecting any of these items you select a part of the total raster picture.

Vector Map

In Vector Maps each entry is represented by its own polygon, as defined in the associated VSF (Vector Specification Format) file. See section “Vector Specification Format” on page 82.

32. Environment Variables

The following subsections describe the GIM resources and environment variables you can change.

Hypertext Browser

M_DOCROOT

This variable is deprecated in version 6.

M_HELPROOT

This variable points to the root of the documentation for the mathematical model. And should be adjusted to the place where you want the documentation to be or where it is for earlier defined help documentation. For local documentation just use th path to the documentation like c:/project/uss/help/ . If you like to refer to an online manual use http://www.yourserver/uss/help

M_HELPSERVER

This variable is deprecated

M_DOCSERVER

If you do not intend to use the standard help viewer provided with the MyM package you should define the name of the http-server where the help-documents are available by setting this variable.

MMM_HOME

If you want the help viewer to come up with a standard help document you should set the *MMM_HOME* variable relative to the pathname of the document to the *M_HELPROOT* environment variable.

M_URL

Is deprecated.

HLPEDITOR

Is deprecated. GIM has it’s own HTML browser included, and does not rely on an external browser anymore.

GIMs Viewer

M_QUICKSHIFT

Interaction on GIM's Model Viewer could be redefined by this variable. The use of this variable is deprecated and replaced by the use of double-click. Double click acts a open/close, and Shift+Double Click zooms a widget in and out.

Part 4 Viewer File Formats

33. Scenario Template Files

For saving and loading complete scenarios the user can define scenario templates A scenario template is an ASCII file where the user specifies which data should be loaded and saved. The template file consists of a series of the following commands:

- HEADER
- RUN
- DIRECTORY
- FILE

Text after the character '!' is comment. The graphical user interface supplies the name entered in the Scenario Manager as an argument to the scenario template which can be used to parameterize the filenames. This parameter is referred to as $1. You can also refer to an environment variable in the scenario template file by $(ENVDEF), where ENVDEF is the enviroment variable.

HEADER

HEADER (cmd)

Defines a standard header for the to data files to be written. 'cmd' is a shell command between double quotes with the same substitution rules as filenames. The HEADER command must precede all FILE commands, and can only be defined once. With the header each data file is forced to get the same header.

RUN

RUN (cmd [, flags]);

Executes the command 'cmd'. 'flags' are optional. Supported flags are:

- "**w**" — execute command only for wrting scenarios
- "**r**" — execute command only for reading scenarios
- "**rw**" — (default) execute command for both reading and writing scenarios

DIRECTORY

DIRECTORY (dirname)
Defines the directory name to be *dirname*. This directory name is used to open all data files. If DIRECTORY is used multiple times then the previous directory name will be overwritten (and thus not added, as with change directory). The default directory for data files is the current directory.

FILE

FILE (filename [, flags]) = varname {, varname} ;
Reads or writes the variables *varname* from file filename . The flags control how en what. The following flags are defined:

- **“r**” — read the file when reading scenarios
- **“w”** — create the file when writing scenarios
- **“a**” — write files in ASCII format
- **“b”** — write files in binary format
- **“l**”: — read data lazy

If no flags are specified, then “rwa” is used as default. If flags are used, r or w should be present at least.
When reading data the type and dimensions of the variable are checked but not the name of the variable! If the l flag is used, only one variable follow.

Lazy read

To save memory only the filenames for lazy variables are read when reading scenarios. Only if the data is really needed (for example in the graphical user interface when a graph should be drawn) these files are read.

34. Vector Specification Format

Widgets can be overlayed with polygons and lines. The Vector Specification Format is ASCII and has the default extension .VSF . This format is also used for Vector Maps.
The syntax of a .VSF file is: *cmd* [*value1 value2*]
The *cmd* is an integer value. The following *cmd*'s are available:

7 set color, using *value1* as an index into the active colormap
8 set x bounds, *value1* defnes the minimum and *value2* the maximum value that x takes.
9 set y bounds, *value1* defines the minimum and *value2* the maximum value for y
10 begin polygon
11 add a point, where *value1* denotes the x value and *value2* the y value
12 end polygon
13 begin line
14 end line
15 ends current drawing object, and resumes drawing at new point (*value1*,*value2*)

Lets examine the following VSF example:

```
7 2
8 0.4 0.9
9 0.6 0.7

10
11 0.5 0.6
11 0.51 0.62
11 0.50 0.63
11 0.51 0.6
12
```

The first line defines the color to be the second entry in the colormap. X -values are defined within [0.4, 0.9] while y values are clipped between [0.6, 0.7]. The first object to be drawn is a polygon (code 10) with {(0.5,0.6), (0.51,0.62), (0.50,0.63), (0.51,0.6)} as the edges of the polygon.

An interesting option is the use of the command “move point” (code 15). This option can be compared with the action “pen up” for plotters. What actually happens is that the draw mode that was started before e.g. drawing a polygon is finished, and the drawing cursor is moved to the new specified point and the drawing mode is resumed. Using the move point option you can define multiple polygons or line to correspond the same data value of a variable. This is especially useful e.g. for coloring countries with islands for the coast, or rivers that are formed by several line parts.

35. Raster Specification File

A raster specification file (extension .RSF) can be used to display a sparse matrix by a raster map widget. A raster map widget is actually an optimized version of certain vector maps. A raster map is built of cells that correspond to geographically data. Each data cell corresponds to a rectangle on the screen. The RSF file should be defined in the File name input field in the Dimension Dialog.
The RSF file has typically the following format:

```
#raster file
2
World -180 180 -90 90
Africa -20 50 -40 40
0.5 0.5
x1 y1
x2 y2
.....
```

36. References

[1] *MyM Language Reference Manual*, 2008.

[2] *MyM Language Tutorial*, 2008.

[3] *MyM Getting Started*, 2008.

37. Index