

Product description

Product name

Deltas Software



Content description

Document type

Deltas
Enabling Delta Life



WTI Ringtoets Framework

User Manual

Version: 1.0
Revision: 1

1 October 2015

WTI Ringtoets Framework, User Manual

Published and printed by:

Deltares
Boussinesqweg 1
2629 HV Delft
P.O. 177
2600 MH Delft
The Netherlands

telephone: +31 88 335 82 73
fax: +31 88 335 85 82
e-mail: info@deltares.nl
www: <https://www.deltares.nl>

For sales contact:

telephone: +31 88 335 81 88
fax: +31 88 335 81 11
e-mail: sales@deltaressystems.nl
www: <http://www.deltaressystems.nl>

For support contact:

telephone: +31 88 335 81 00
fax: +31 88 335 81 11
e-mail: support@deltaressystems.nl
www: <http://www.deltaressystems.nl>

Copyright © 2015 Deltares

All rights reserved. No part of this document may be reproduced in any form by print, photo print, photo copy, microfilm or any other means, without written permission from the publisher: Deltares.

Contents

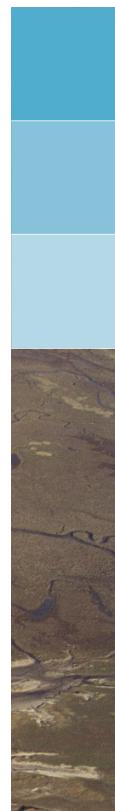
1 A guide to this manual	1
1.1 Introduction	1
1.2 Overview	1
1.3 Manual version and revisions	1
1.4 Typographical conventions	1
1.5 Changes with respect to previous versions	2
2 Background	3
2.1 About the framework	3
2.2 Framework: Vision Statement	5
2.3 Overall Architecture	5
3 General overview of the GUI	7
3.1 Windows	7
3.1.1 Project	8
3.1.2 Main (central) window	9
3.1.3 Map	10
3.1.4 Data	11
3.1.5 Chart	12
3.1.6 Properties	12
3.1.7 Messages	13
3.2 Dockable views	13
3.2.1 Docking tabs separately	13
3.2.2 Multiple tabs	14
3.3 Context menus	16
3.3.1 Project	16
3.3.1.1 Project level	16
3.3.1.2 Other level still to come	17
3.3.1.3 Other items	17
3.3.2 Main (Central Map window)	17
3.3.2.1 Table editor	18
3.3.2.2 Charts	18
3.3.3 Map	18
3.3.4 Messages	19
3.4 Ribbons and toolbars	19
3.4.1 Ribbons (hot keys)	19
3.4.2 File	20
3.4.3 Home	21
3.4.4 View	21
3.4.5 Tools	22
3.4.6 Map	22
3.4.7 Scripting	23
3.4.8 Shortcuts	24
3.4.9 Quick access toolbar	24

List of Figures

2.1	NEEDS TO BE UPDATED / REMOVED: Mindmap with key user stories.	3
2.2	The structure of DeltaShell as integrated modelling suite.	4
2.3	TO BE UPDATED: From real world to model results.	5
2.4	TO BE UPDATED: Mindmap with key architecture subjects of Delta Shell.	6
3.1	Overview of the graphical user interface, example for a SOBEK3 model.	7
3.2	The project tree window.	8
3.3	The central map view.	9
3.4	The chart window view.	10
3.5	The map window.	11
3.6	The data window.	11
3.7	Example of the chart window.	12
3.8	Example of a property grid in the properties window of a flow model.	12
3.9	The messages window.	13
3.10	The message detail window.	13
3.11	Docking windows on two screens within the framework.	14
3.12	Bringing the Undo/Redo window to the front	14
3.13	Docking the Undo/Redo window.	15
3.14	Auto hide the Undo / Redo window	16
3.15	The context menu on the project level within the project explorer.	17
3.16	The context menu of the table editor.	18
3.17	The context menu of the chart view.	18
3.18	The context menu for the Messages window.	19
3.19	Perform operations using the hot keys	19
3.20	The <i>File</i> ribbon.	20
3.21	The options dialog.	21
3.22	The <i>Home</i> ribbon.	21
3.23	The <i>View</i> ribbon.	21
3.24	The <i>Tools</i> ribbon.	22
3.25	The <i>Map</i> ribbon.	22
3.26	The ribbon with minimized categories.	23
3.27	The scripting <i>ribbon</i> within deltashell.	23
3.28	The quick access toolbar.	25

List of Tables

3.1	Functions and their descriptions within the scripting <i>ribbon</i> of deltashell.	23
3.1	Functions and their descriptions within the scripting <i>ribbon</i> of deltashell.	24
3.2	Shortcut keys within the scripting editor of deltashell.	24



1 A guide to this manual

1.1 Introduction

This User Manual concerns the framework.

1.2 Overview

To make this manual more accessible, we will briefly describe the contents of each chapter.

[Chapter 2: Background](#), introduces the framework as an integrated modeling environment to provide users with a single application which acts as a platform to integrate various calculations and tools.

[Chapter 3: General overview of the GUI](#), gives a brief introduction to all GUI-components, which are shared between fail mechanisms based on the framework.

1.3 Manual version and revisions

This manual applies to WTI Ringtoets (version 2015).

1.4 Typographical conventions

Throughout this manual, the following conventions help you to distinguish between different elements of text.

Example	Description
Piping Scenarios	Title of a window or sub-window. Sub-windows are displayed in the Module window and cannot be moved. Windows can be moved independently from the Module window, such as the Visualisation Area window.
Save	Item from a menu, title of a push button or name of a user interface input field. Upon selecting this item (click or, in some cases, double click with the left mouse button on it) a related action will be executed; in most cases it will result in displaying some other (sub-)window. In case of an input field, you are supposed to enter input data of the required format and in the required domain.
<\tutorial\DR10\asfalt> <revetments.csv.mdw>	Directory names, filenames, and path names are expressed between angle brackets, <>.
“27 08 1999”	Data to be typed by you into the input fields are displayed between double quotes. Selections of menu items, option boxes etc. are described as such: for instance ‘select Save and go to the next window’.

Example	Description
wti-menu	Commands to be typed by you are given in the font Courier New, 10 points.
	User actions are indicated with this arrow.
[m/s] [-]	Units are given between square brackets when used next to the formulae. Leaving them out might result in misinterpretation.

1.5 Changes with respect to previous versions

This is the first edition.

2 Background

2.1 About the framework

The framework will be gradually extended to allow for the integration of a variety of environmental models.

The framework is an integrated modeling environment to provide users with a single application which acts as a platform to integrate various locations and failure mechanisms. This is achieved by making use of a software framework specifically focused to provide a set of components which can be reused by all kinds of models. The overall picture of what functionality needs to be provided has been proposed by a variety of (potential) users and has been captured in so-called user stories. These user stories have been gathered in a so-called mind map which is presented in [Figure 2.1](#). As can be seen from the mind map, the framework

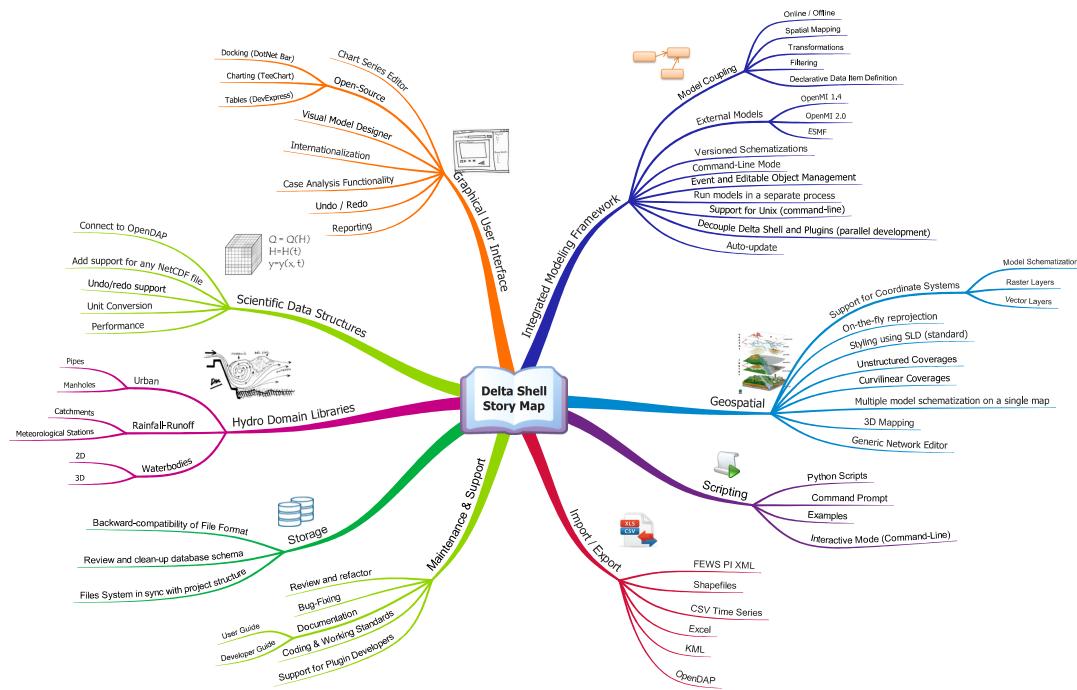


Figure 2.1: NEEDS TO BE UPDATED / REMOVED: Mindmap with key user stories.

has to comply with a broad range of wishes defined by various users and user groups. Main topics (depicted in the main branches) are:

- ◊ The definition of the integrated modeling framework,
- ◊ Data management: structuring, importing/exporting, storage.
- ◊ Dealing with environmental domain libraries (hydro, geo).
- ◊ Application interfaces (APIs) such as a graphical user interface or a command line interface using scripts for batch operation.
- ◊ Documentation and support

The framework provides a user-friendly and open environment for different locations and failure mechanisms. [Figure 2.2](#) shows the principles of the framework. It allows to combine different failure mechanisms (plug-ins), such as piping, asphalt revetments, or macrostability. In this fashion, it is possible to compose various dedicated software suites within a single framework while preserving the same look-and-feel at the same time. Furthermore, common

plug-ins with generic functionality may be used by all model plug-ins that have been integrated within the framework. Finally, tools for setting up or importing different types of data, perform calculations of the different failure mechanisms or combinations of them, and analyse model results.

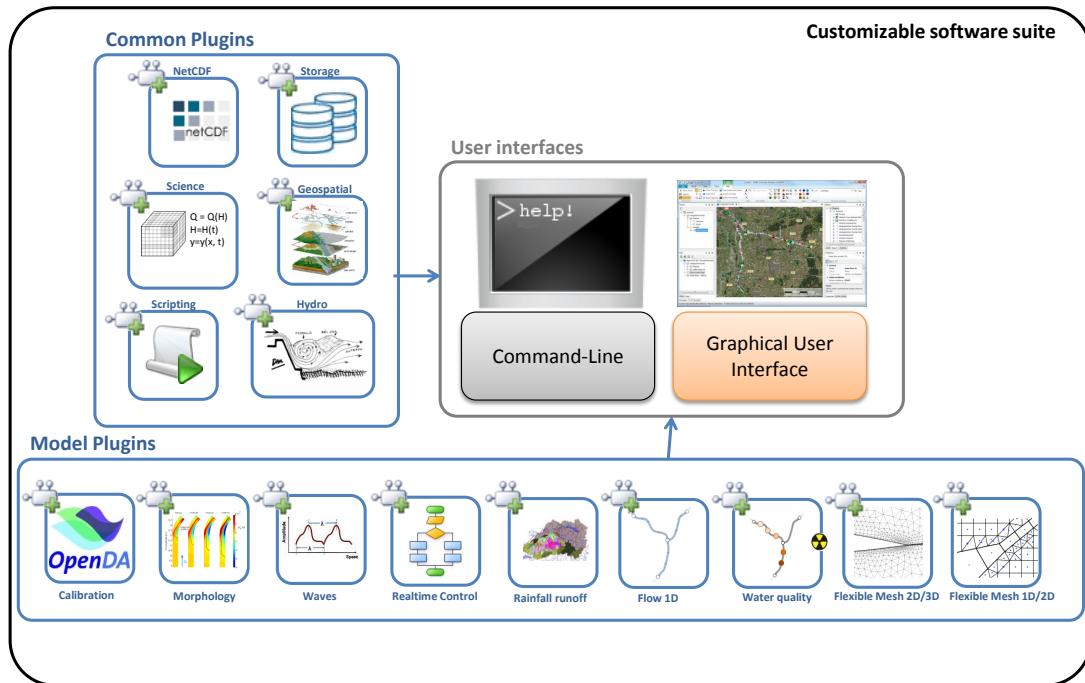


Figure 2.2: The structure of DeltaShell as integrated modelling suite.

A model is, by definition, a simplified version of (a possible) reality. A generic model is a mathematical description of physical processes provided as a computer program. In combination with data for a specific part of the world, the model becomes a site-specific model. Within a site-specific model a set of boundary conditions forms a scenario.

The framework follows a layered concept to get from 'real world' to 'numerical model result' ([Figure 2.3](#)). Real objects usually are available on maps or other (digital) format. Based on such data, the framework helps to create a schematisation, i.e. a network with model objects that correspond to the real objects. The schematisation makes the model site-specific. This model can be run under different sets of boundary conditions (scenarios). Numerical solution of the equations with a failure mechanism plugin finally produces model results.

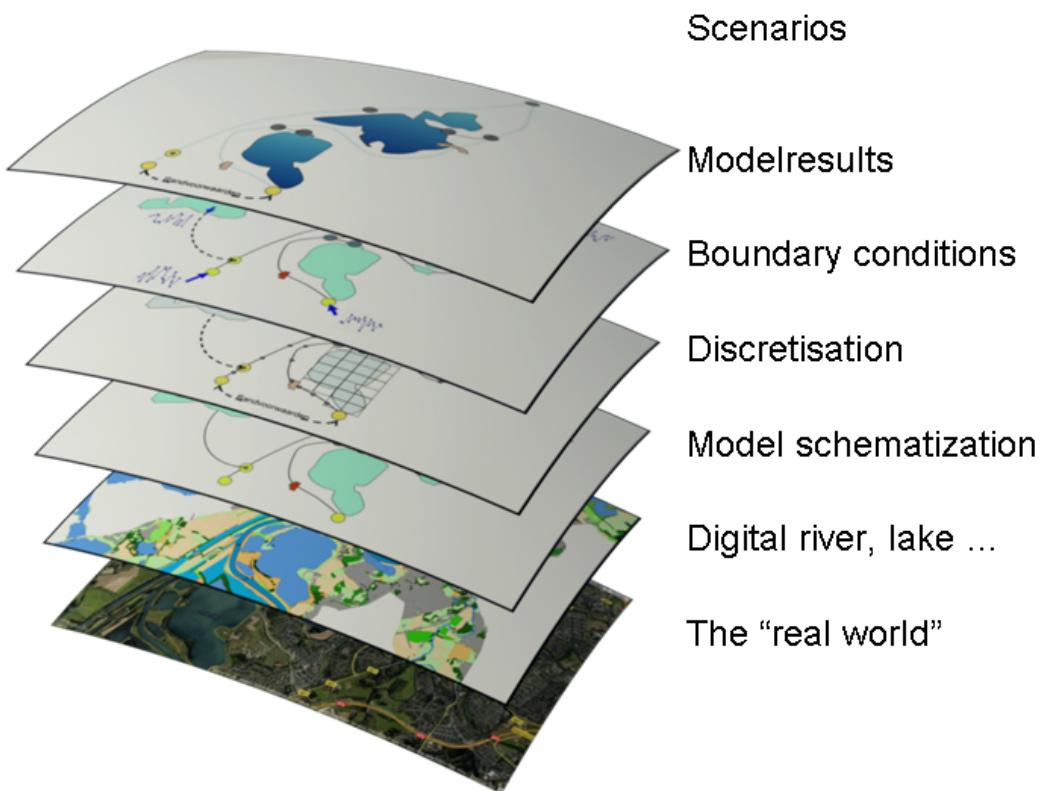


Figure 2.3: TO BE UPDATED: From real world to model results.

2.2 Framework: Vision Statement

For users:

The framework should facilitate modelers in: schematization (with/without user interface), running tests in standardized modelling environments, batch execution (from command line) and parallel execution (on clusters). All modellers should be able to use the Framework combined with the failure mechanisms of their preference (piping, macrostability, etc.).

For developers:

The Framework should provide a generic, transparent and simple way of developing new or combining existing failure mechanisms (plug-ins). The emphasis should be on reuse and expansion of existing code and features.

2.3 Overall Architecture

This section gives an overview of the framework architecture and a description of its main components. The framework has been designed using a flexible architecture which can easily facilitate the use of external applications. The concept of the overall architecture is similar to existing commercial products like Eclipse or Microsoft Visual Studio Shell. The most important subjects considering the architecture of Delta Shell are presented in [Figure 2.4](#). These are the subjects that need to come together within the architecture of the framework.

TO BE CONTINUED

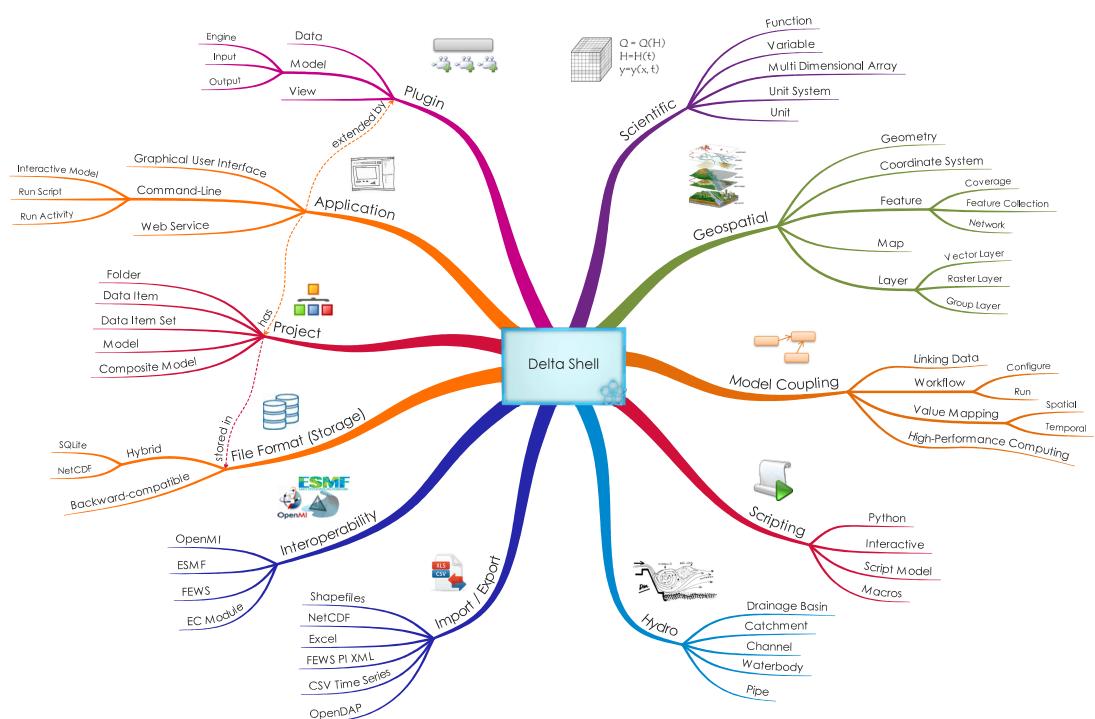


Figure 2.4: TO BE UPDATED: Mindmap with key architecture subjects of Delta Shell.

3 General overview of the GUI

As shown in ??, one of the interfaces within the framework makes it possible to extend the available (model) plug-ins with Graphical User Interfaces (GUIs). In this fashion, all user interfaces of model plug-ins integrated have the same look-and-feel.

This chapter introduces all GUI-components, which are shared between applications based on the framework.

3.1 Windows

As DeltaShell is an integrated modelling suite, the application is project-based. Within a project several models may be run and combined.

The main user interface is organized in a set of tool and document windows. An example is given in Figure 3.1. The tool windows show properties of the current project, whereas document windows are used to visualize or edit a specific data type. Tool windows can be docked where you prefer — even at a second display. Document windows are, when placed within the framework, always in the central area but may also be docked stand-alone (on a second display, for example). Examples of tool windows are:

- ◊ Project
- ◊ Map
- ◊ Properties
- ◊ Chart
- ◊ Messages

Examples of document windows are:

- ◊ Map(s)
- ◊ Editor(s)

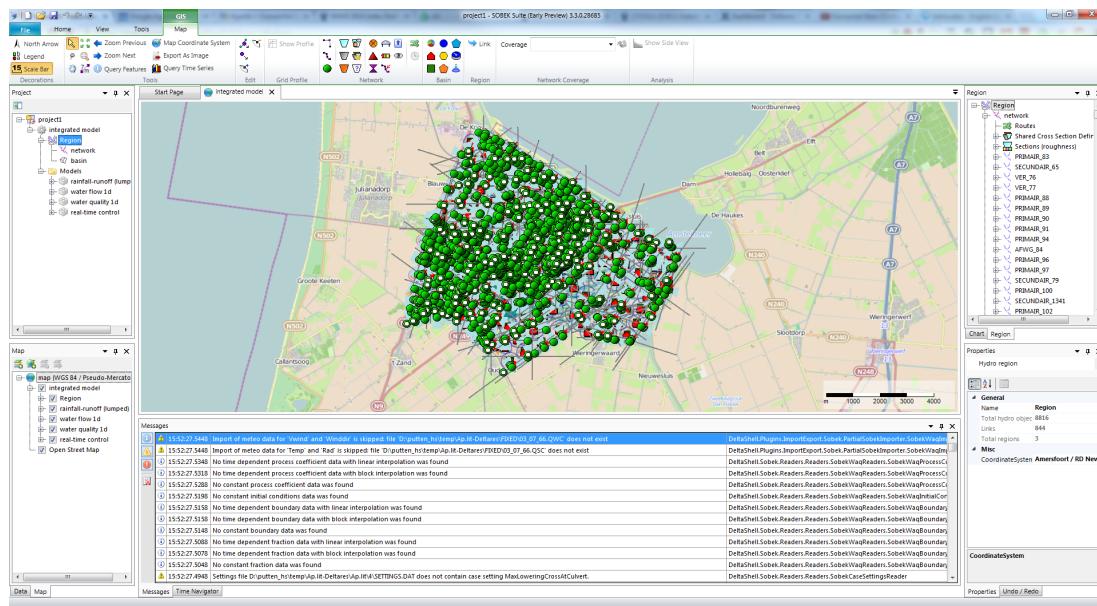


Figure 3.1: Overview of the graphical user interface, example for a SOBEK3 model.

In this chapter all tool windows, menus, dockable views, context menus, and ribbons and

toolbars will be described. Map functionality and the spatial editor are treated in separate chapters. The specific editors for the different models are described in the user manuals belonging to those model plug-ins.

3.1.1 Project

The **Project** window is the main navigation window for the project data, showing the total workspace in a tree view ([Figure 3.2](#)). In the **Project** all project components are shown. All project items with sub-levels can be collapsed by a mouse-click on the ‘–’ sign in the tree view. Project data can be sorted by adding new folders to the project tree view and moving models or movable items to designated folders. By clicking on the top left icon in [Figure 3.2](#) the active item in the central Map is located in the tree view.

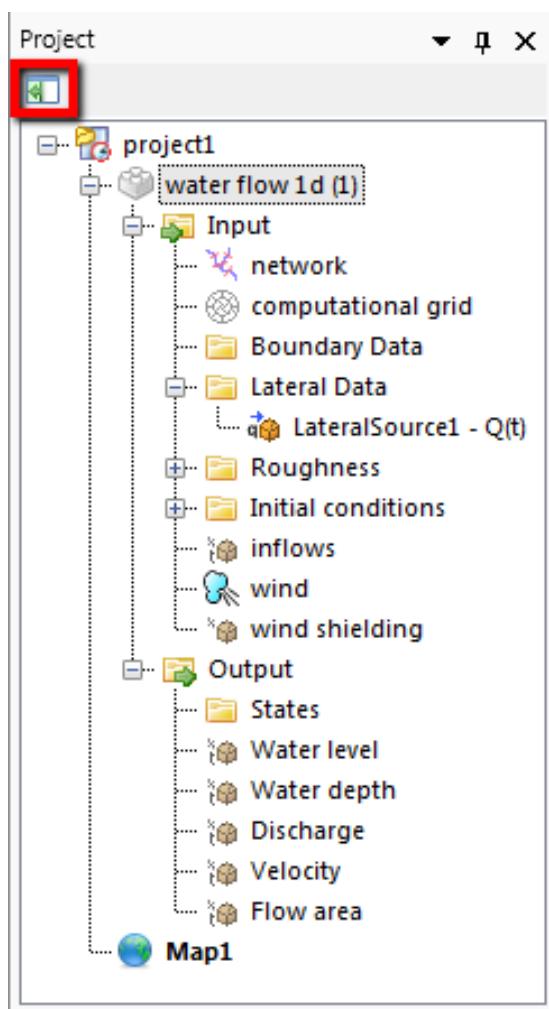


Figure 3.2: The project tree window.

Several possibilities exist to work with the tree view:

- ◊ Left mouse-click to select
- ◊ Right mouse-click gives a context menu with available actions
- ◊ Double-click to show a map or editor in the main (central) window, depending on the parameter

3.1.2 Main (central) window

The main window ([Figure 3.3](#)) is by default always placed in the middle of the screen. It can also be docked separately, for example on a second display. It is used to present a map for all geo-referenced modeldata, the editors for other data, and results in charts. The editors for other data are model-specific and therefore described in the manuals for the various model plug-ins.

All items with a geo-reference will be presented on the central map, for example: network, computational grid and output data as layers, comparable to a geographical information system (GIS). Working with these layers is described in [??](#).

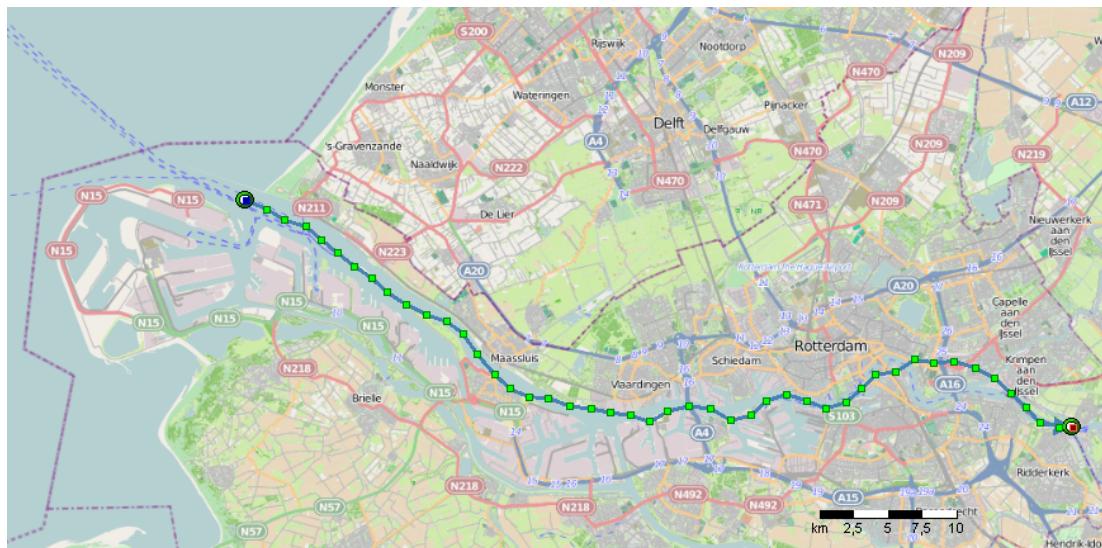
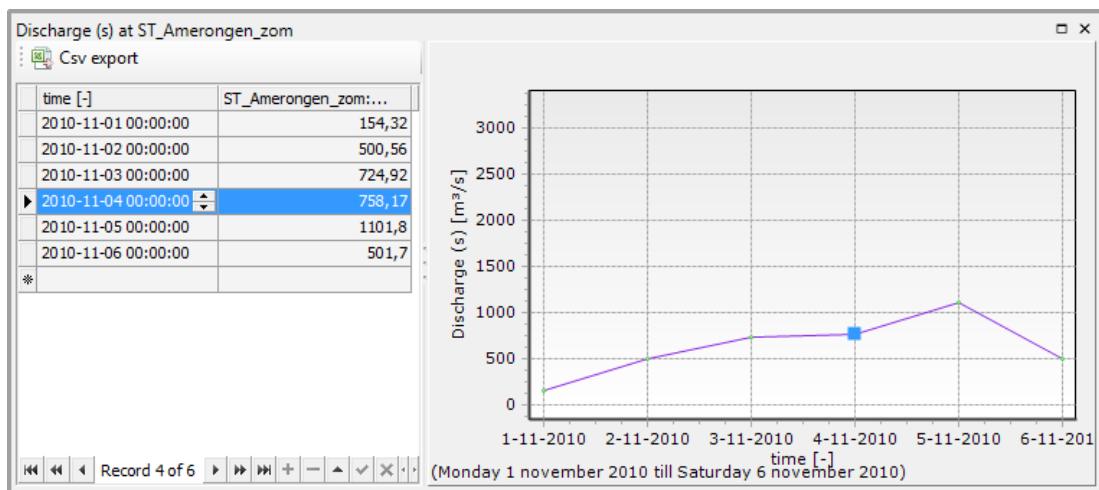


Figure 3.3: The central map view.

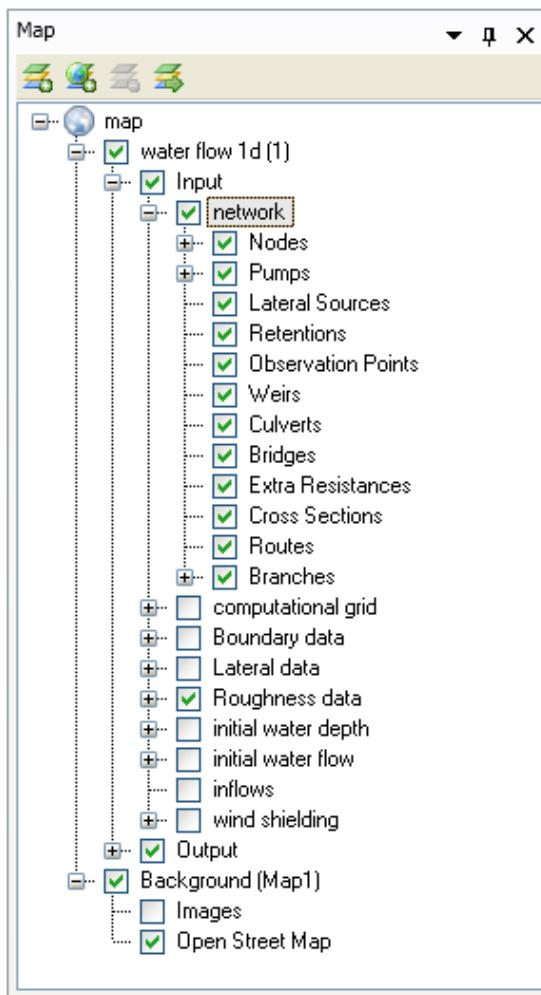
When working in the central map, for example on a network, it is possible to add, adjust or delete network components, which is described thoroughly in the dflow manual.

Results in charts windows ([Figure 3.7](#)) are presented by combining a table view and a '*xy*'-plot. The user can visualize separate data points by selecting rows in the table view. The data that is presented in the chart window may also be exported to a *.csv file by clicking the *Csv export* button.

**Figure 3.4:** The chart window view.

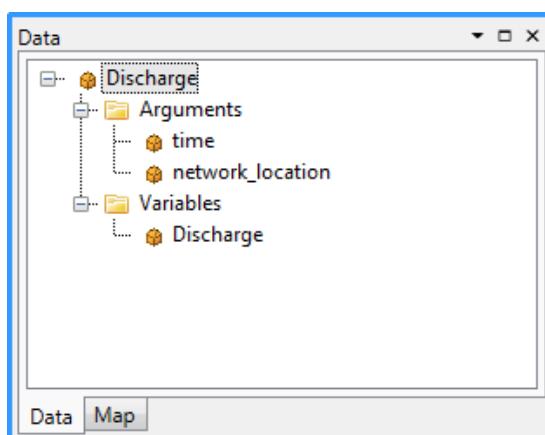
3.1.3 Map

The **Map** window (Figure 3.5) manages the active map. In this window layers within the active map can be shown, hidden or adjusted. With the four icons in the top left of the window new <shp>- or <wms>-layers can be added, removed or exported. With the icons in the top right of the window, the window can be removed or hidden. The window can be retrieved by clicking on *Map* in *View* ribbon.

**Figure 3.5:** The map window.

3.1.4 Data

The **Data** window (Figure 3.6) can be used to inspect the contents of available data items, when selected in the project window.

**Figure 3.6:** The data window.

3.1.5 Chart

The **Chart** window (Figure 3.7) can be used to stack/unstack chart series of the same type or to convert chart series to a certain type (area, line, point, or bar series). Furthermore, series can be selected or deselected within this window.

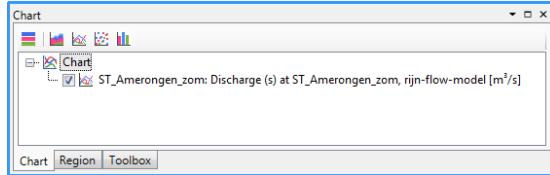


Figure 3.7: Example of the chart window.

3.1.6 Properties

The **Properties** window shows properties for an active selection of the graphical user interface. When a model object is selected in the **Region** window it shows the properties of this object. Accordingly, the **Properties** window of an item selected in the **Project** shows data related to the selected item, for example the simulation time of a <flow model> or a list with output parameters when clicking on the <output> entry. Figure 3.8 shows an example for the properties of a flow model.

In the **Properties Window** data can also be edited. If the property grid is insufficient to display the information, for example in case of time series, an additional editor can be opened.

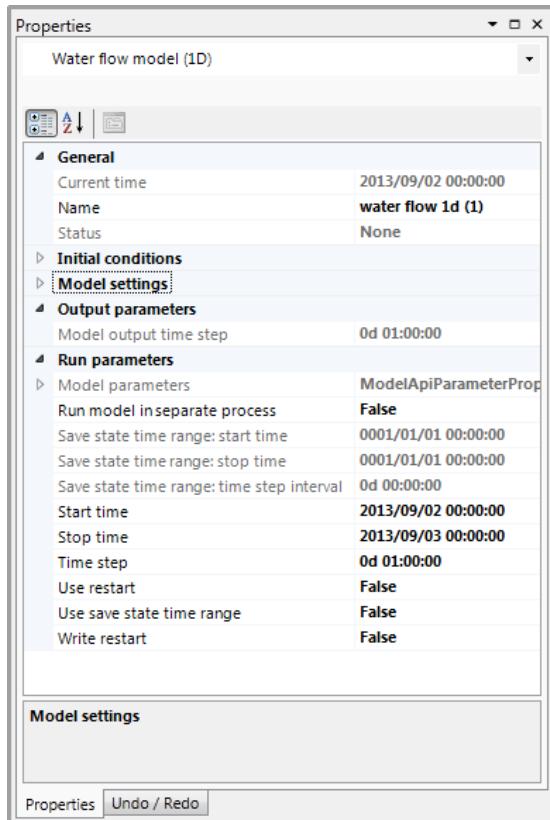


Figure 3.8: Example of a property grid in the properties window of a flow model.

3.1.7 Messages

The **Messages** window (Figure 3.9) is a logging window. Messages sent from models or different parts of the system are shown here. When a message is too large to fit within the **Messages** the user can open a single message (Figure 3.10) separately by right-mouse-clicking the message and selecting the 'Show details' option.

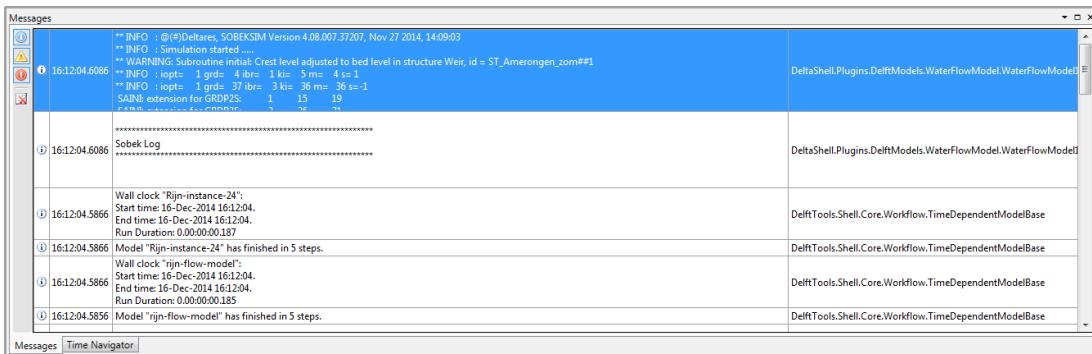


Figure 3.9: The messages window.

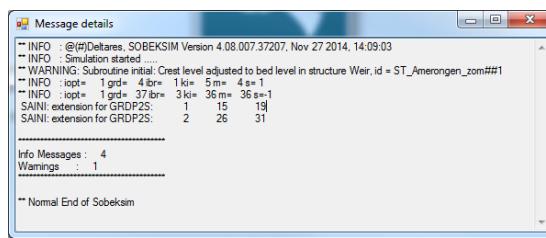


Figure 3.10: The message detail window.

Within the **Messages** window the user may select the verbosity of the shown messages, ranging from 'Info messages' to 'Warning messages' to 'Error messages'. It is also possible to clear all messages by clicking .

Furthermore, a run report is shown in the output in the **Project** for each model simulation. This run report contains all the messages (from DeltaShell and the model plug-ins) that occur during a simulation.

Finally, an application log is kept for each session of DeltaShell in the project database. In this log-file, which can be accessed through the *File/Help* or *Home* menus, all messages are stored.

3.2 Dockable views

The framework offers lots of freedom to customize dockable views, which are discussed in this section.

3.2.1 Docking tabs separately

Within the framework the user can dock the separate windows according to personal preferences. These preferences are then saved for future use of the framework. An example of such preferences is presented in Figure 3.11, where windows have been docked on two screens.

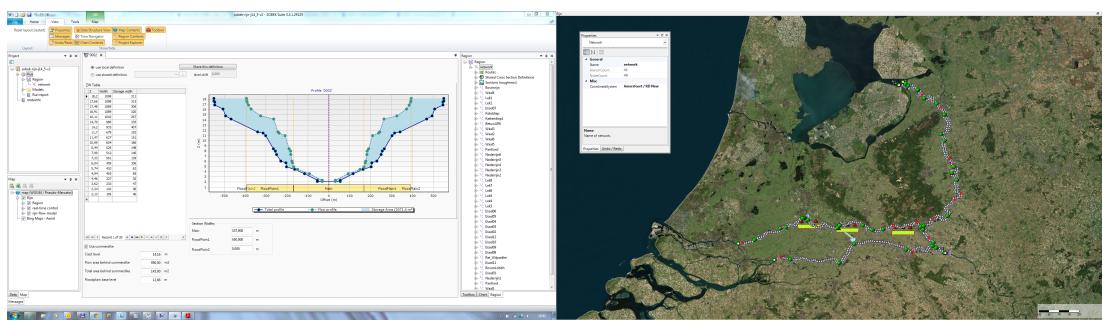


Figure 3.11: Docking windows on two screens within the framework.

3.2.2 Multiple tabs

In case two windows are docked in one view, the underlying window (tab) can be brought to the front by simply selecting the tab, as is shown here.

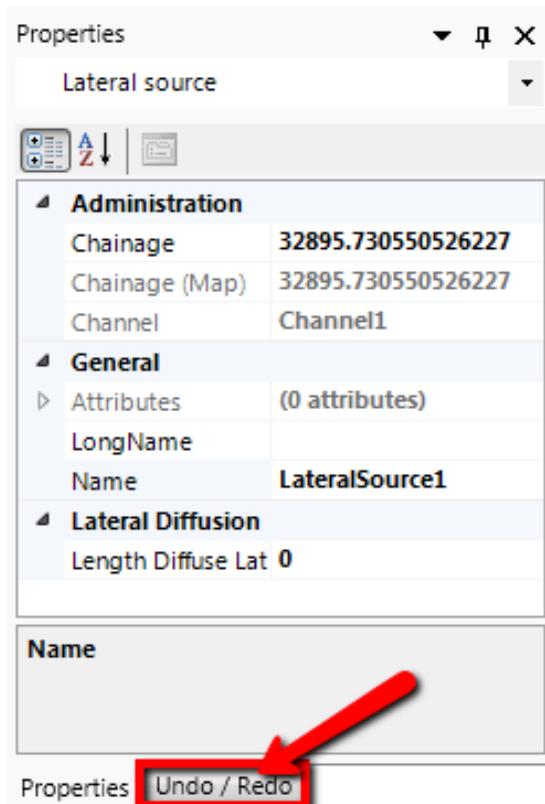


Figure 3.12: Bringing the **Undo/Redo** window to the front

By dragging dockable windows with the left mouse button and dropping the window left, right, above or below another one the graphical user interface can be customized according to personal preferences. Here an example of the **Undo/Redo** window being docked above the **Properties** window.

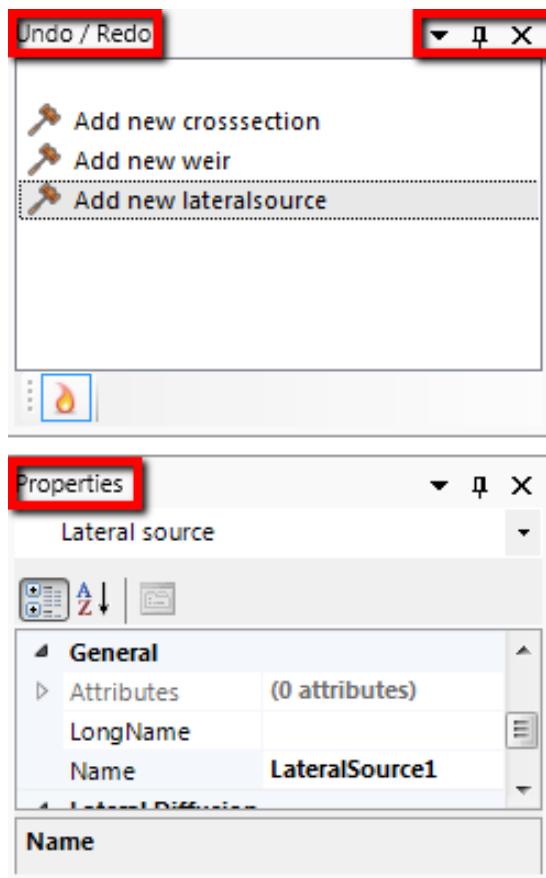


Figure 3.13: Docking the **Undo/Redo** window.

Additional features are the possibility to remove or (auto) hide the window (top right in [Figure 3.13](#)). In case of removal, the window can be retrieved by a mouse-click on *Undo/Redo* in the *View* ribbon. Hiding the **Undo/Redo** window results in:

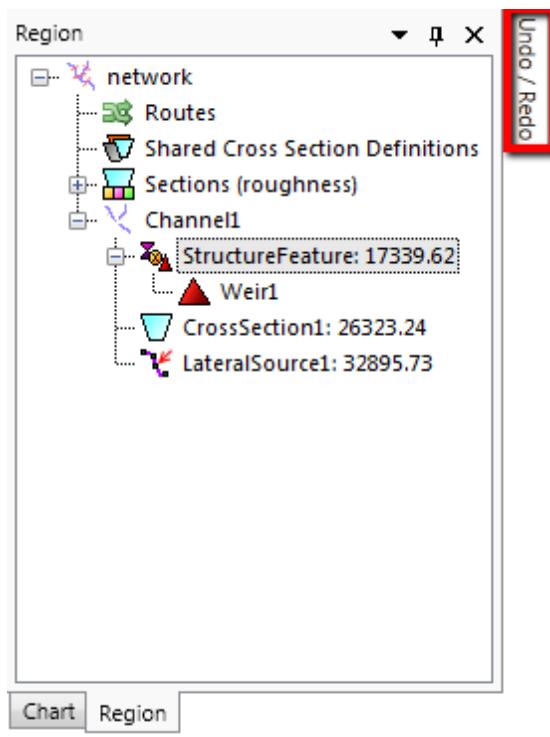


Figure 3.14: Auto hide the **Undo / Redo** window

3.3 Context menus

Depending on the active window, different context menus are present when right-mouse-clicking on items within this window. This section will treat these context menus per active window.

3.3.1 Project

Within the **Project** window, a variety of levels and/or items with different context menus are present. These will be described here.

3.3.1.1 Project level

The context menu of the project level within the project explorer is shown in [Figure 3.15](#). It contains the following choices:

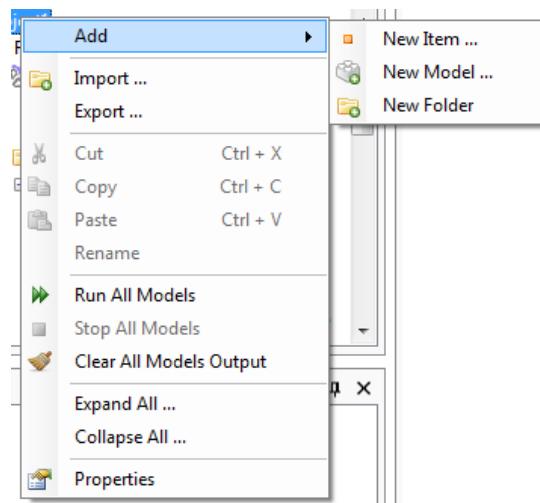


Figure 3.15: The context menu on the project level within the project explorer.

- ◊ *Add* ⇒ Option to add models and/or items to project (depending on installed model plug-ins)
- ◊ *Import ...* ⇒ Opens selection window for a variety of available importers (if present)
- ◊ *Export ...* ⇒ Opens selection window for a variety of available exporters (if present)
- ◊ *Cut* ⇒ Cuts current project for pasting elsewhere
- ◊ *Copy* ⇒ Copies current project for pasting elsewhere
- ◊ *Paste* ⇒ Pastes current project available on the clipboard
- ◊ *Rename* ⇒ Rename the current project
- ◊ *Run All Models* ⇒ Runs all models available in the project
- ◊ *Stop All Models* ⇒ Stops running of all models currently running within the project
- ◊ *Clear All Models Output* ⇒ Clears all model output of models available within the project
- ◊ *Expand All ...* ⇒ Expands all project items
- ◊ *Collapse All ...* ⇒ Collapses all project items
- ◊ *Properties* ⇒ Switches to **Properties** window of active project

3.3.1.2 Other level still to come

The context menu of ...

3.3.1.3 Other items

For the other items in the project explorer the following holds:

- ◊ Folder items have the same context menu as the project level context menu depicted in [Figure 3.15](#).
- ◊ All other project items have context menus similar to the context menu on the region level as shown in ??

3.3.2 Main (Central Map window)

The **Main** or **Central Map** window may consist of multiple tabs ...

3.3.2.1 Table editor

The context menu of the table editors within the **Main** window as shown in [Figure 3.16](#) contains the following choices:

A screenshot of a table editor showing a context menu. The table has columns for Time [yyyy-MM-dd ...] and flow... (with values 500, 500, 500, 500, 500). A context menu is open over the third row, listing 'Copy', 'Paste', and 'Delete' options.

Time [yyyy-MM-dd ...]	flow...
2010-11-01 00:00:00	500
2010-11-02 23:59:00	500
2010-11-03 00:00:00	500
2010-11-04 00:00:00	500
2010-11-04 00:01:00	500
2010-11-06 00:00:00	500
*	

Figure 3.16: The context menu of the table editor.

- ◊ *Copy*
- ◊ *Paste*
- ◊ *Delete*

3.3.2.2 Charts

The context menu of the charts within the **Main** window as shown in [Figure 3.17](#) contains the following choice:

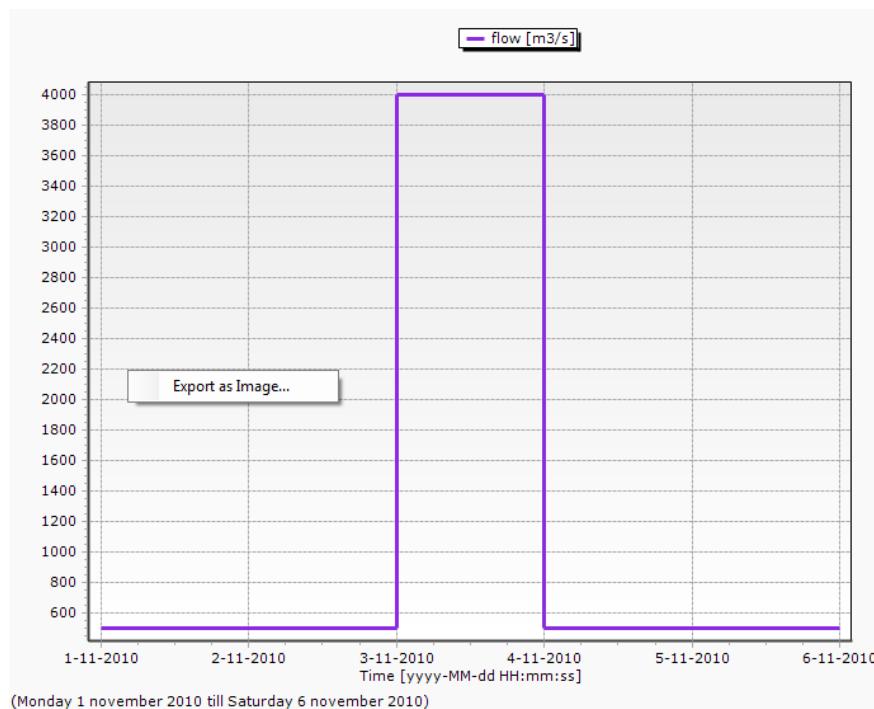


Figure 3.17: The context menu of the chart view.

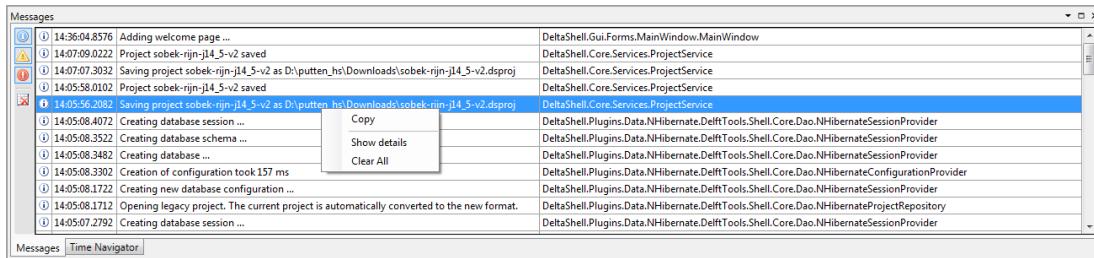
- ◊ *Export as Image ...*

3.3.3 Map

The context menus for the **Map** window are described in [??](#).

3.3.4 Messages

The context menu for the **Messages** window as shown in [Figure 3.18](#) contains the following choices:



[Figure 3.18: The context menu for the **Messages** window.](#)

- ◊ *Copy*
- ◊ *Show details* ⇒ Open separate window with detailed message information
- ◊ *Clear all*

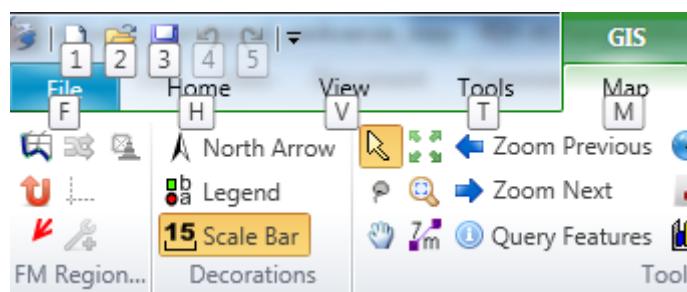
3.4 Ribbons and toolbars

The user can access the toolbars arranged in *ribbons*. Failure mechanisms plug-ins can have their own specific *ribbon*. The *ribbon* may be auto collapsed by activating the *Collapse the Ribbon* button when right-mouse-clicking on the *ribbon*.

3.4.1 Ribbons (hot keys)

WTI Ringtoets makes use of ribbons, just like Microsoft Office. You can use these ribbons for most of the operations. With the ribbons comes hot key functionality, providing shortcuts to perform operations. If you press “ALT”, you will see the letters and numbers to access the ribbons and the ribbon contents (i.e. operations). For example, “ALT” + “H” will lead you to the “Home”-ribbon ([Figure 3.19](#)).

Note: Implementation of the hot key functionality is still work in progress.



[Figure 3.19: Perform operations using the hot keys](#)

3.4.2 File

The left-most *ribbon* is the *File* ribbon. It has menu-items comparable to most Microsoft applications. Furthermore, it offers users import and export functionality, as well as the *Help* and *Options* dialogs, as shown in [Figure 3.20](#) and [Figure 3.21](#).

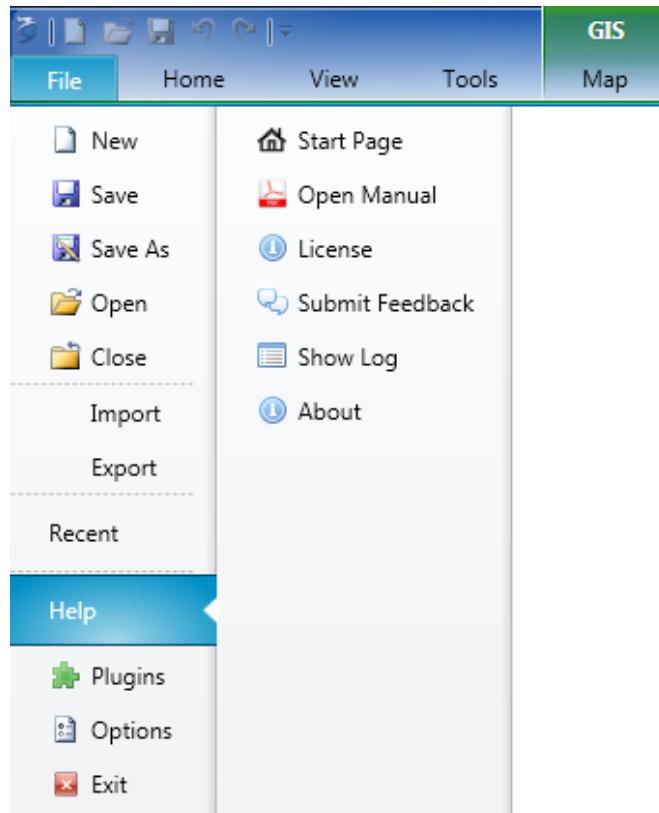
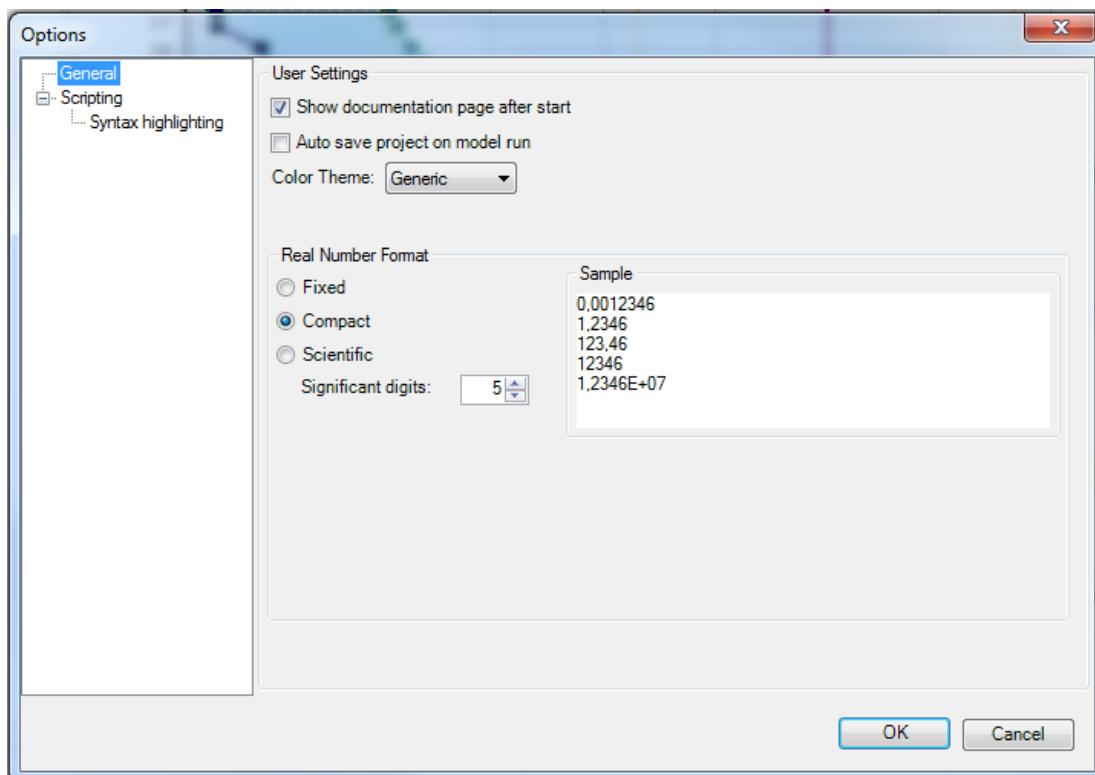
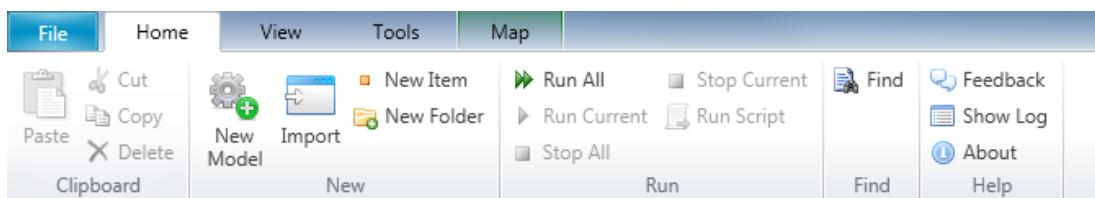


Figure 3.20: The File ribbon.

**Figure 3.21:** The options dialog.

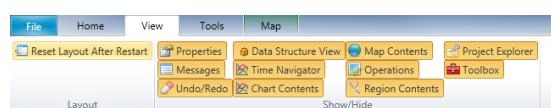
3.4.3 Home

The second *ribbon* is the *Home* ribbon (Figure 3.22). It harbours some general features for clipboard actions, addition of items, running models, finding items within projects or views, and help functionality.

**Figure 3.22:** The Home ribbon.

3.4.4 View

The third *ribbon* is the *View* ribbon (Figure 3.23). Here, the user can show or hide windows.

**Figure 3.23:** The View ribbon.

3.4.5 Tools

The fourth *ribbon* is the *Tools* ribbon (Figure 3.24). By default, it contains only the *Open Case Analysis View* tool. Some model plug-ins offer the installation of extra tools that may be installed. These are documented within the user documentation of those model plug-ins.

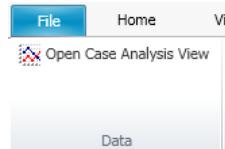


Figure 3.24: The Tools ribbon.

3.4.6 Map

The last *ribbon* is the *Map* ribbon (Figure 3.25).

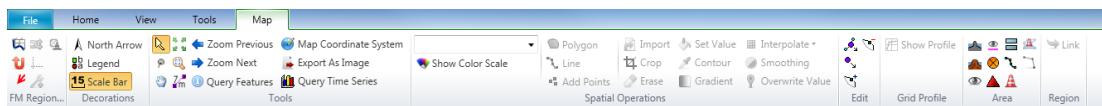


Figure 3.25: The Map ribbon.

This will be used heavily, while it harbours all *Geospatial* functions, like:

- ◊ *Decorations* for the map
 - North arrow
 - Scale bar
 - Legend
 - ...
- ◊ *Tools* to customize the map view
 - Select a single item
 - Select multiple items by drawing a curve
 - Pan
 - Zoom to Extents
 - Zoom by drawing a rectangle
 - Zoom to Measure distance
 - ...
- ◊ *Edit* polygons, for example within a network, basin, or waterbody
 - Move geometry point(s)
 - Add geometry point(s)
 - Remove geometry point(s)
- ◊ Creation of a model *Network*, for example for D-Flow 1D
 - Add new Branch
 - Split Branch
 - Add Cross section
 - Add Weir
 - Add Pump
 - ...



Note: The *ribbons* adjust to the size of the application window. If, for what reason, the user wants to minimize the window, the ribbons might look like as shown in [Figure 3.26](#). Some of the *ribbon* categories have been condensed into a single drop-down panel.

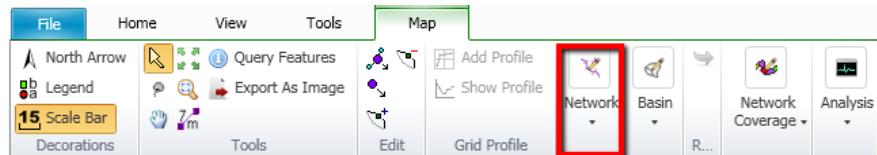


Figure 3.26: The ribbon with minimized categories.

Still, all functions of the category can be activated as they will appear in the drop-down panel.

3.4.7 Scripting

When you open the scripting editor in deltashell, a Scripting *ribbon* category will appear. This ribbon has the following additional options (see also [Figure 3.27](#)), which are described in Table 3.1:



Figure 3.27: The scripting ribbon within deltashell.

Table 3.1: Functions and their descriptions within the scripting ribbon of deltashell.

Function	Description
Run script	Executes the selected text. If no text is selected then it will execute the entire script
Clear cached variables	Clears all variables and loaded libraries from memory
Debugging	Enables/Disables the debug option. When enabled you can add breakpoint to the code (using F9 or clicking in the margin) and the code will stop at this point before executing the statement (use F10 (step over) or F11 (step into) for a more step by step approach)
Python variables	Show or hide python variables (like _var_) in code completion
Insert spaces/tabs	Determines if spaces or tab characters are added when pressing tab
Tab size	Sets the number of spaces that are considered equal to a tab character
Save before run	Saves the changes to the file before running
Create region	Creates a new region surrounding the selected text
Comment selection	Comments out the selected text
Convert to space indenting	Converts all tab characters in the script to spaces. The number of spaces is determined by Tab size
Convert to tab indenting	Converts all x number of space characters (determined by Tab size) in the script to tabs
Python (documentation)	Opens a link to the python website, showing you the python syntax and standard libraries

Table 3.1: Functions and their descriptions within the scripting ribbon of deltashell.

Function	Description
deltashell (documentation)	Opens a link to the deltashell documentation website (generated documentation of the deltashell api)

3.4.8 Shortcuts

The shortcut keys of the scripting editor within deltashell are documented in Table 3.2.

Table 3.2: Shortcut keys within the scripting editor of deltashell.

Shortcut	Function
Ctrl + Enter	Run selection (or entire script with no selection)
Ctrl + Shift + Enter	Run current region (region where the cursor is in)
Ctrl + X	Cut selection
Ctrl + C	Copy selection
Ctrl + V	Paste selection
Ctrl + S	Save script
Ctrl + -	Collapse all regions
Ctrl + +	Expand all regions
Ctrl + "	Comment or Uncomment current selection
Ctrl + W	Add selection as watch variable
Ctrl + H	Highlight current selection in script (press esc to cancel)
F9	Add/remove breakpoint (In debug mode only)
F5	Continue running (In debug mode only — when on breakpoint)
Shift + F5	Stop running (In debug mode only — when on breakpoint)
F10	Step over current line and break on next line (In debug mode only - when on breakpoint)
F11	Step into current line if possible, otherwise go to next line (In debug mode only — when on breakpoint). This is used to debug functions declared in the same script (that have already runned)

3.4.9 Quick access toolbar



Note: The user can make frequently used functions available by a single mouse-click in the *Quick Access Toolbar*, the top-most part of the application-window. Do this by right-mouse-clicking a ribbon item and selecting *Add to Quick Access Toolbar*.

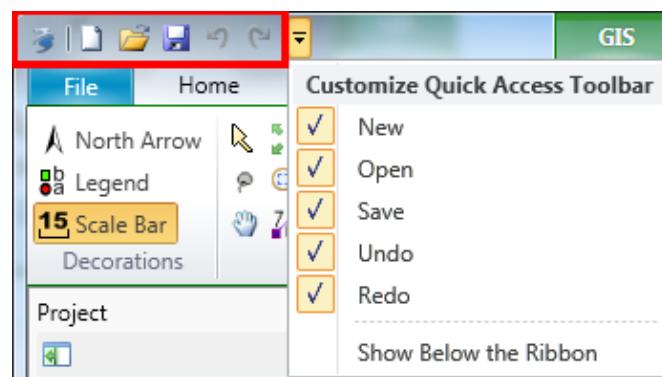
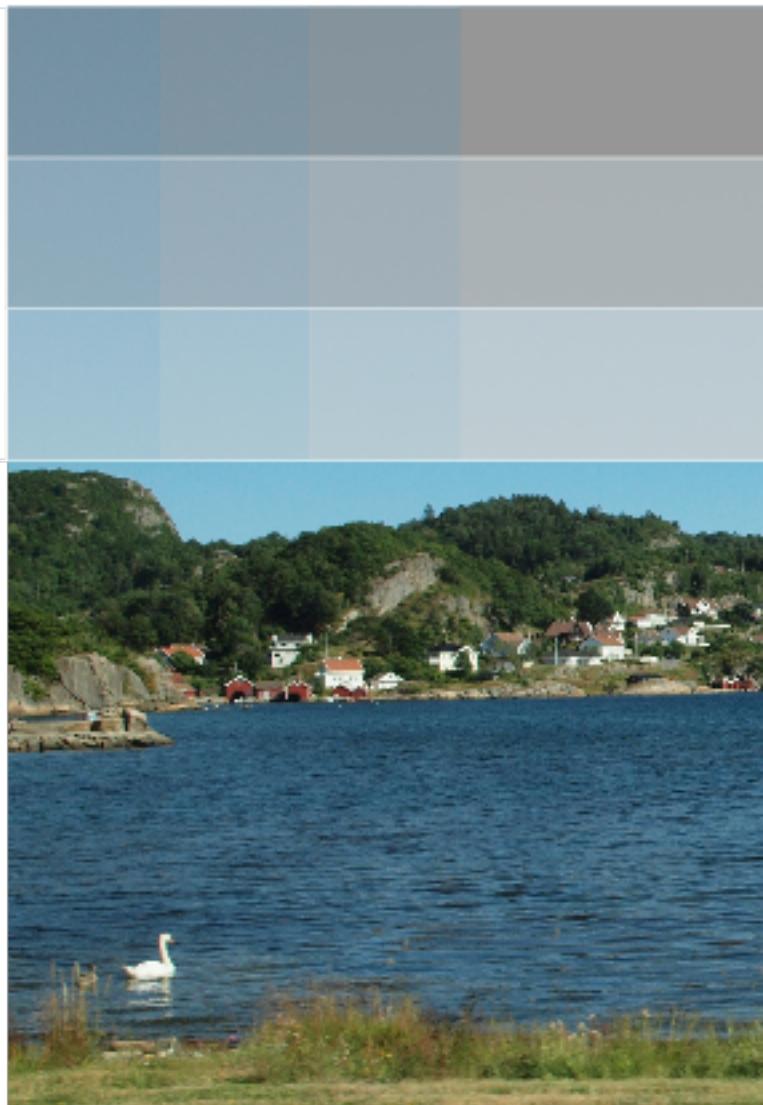


Figure 3.28: The quick access toolbar.



Deltarès systems

P.O. Box 177
2600 MH Delft
Boussinesqweg 1
2629 HD Delft
The Netherlands

+31 (0)88335 8188
sales@deltaressystems.nl
www.deltaressystems.nl