

Documentation
of
FullSWOF_UI

v2.00.00 (2020-06-12)

Compatible with
FullSWOF_1D 1.02.02
and
FullSWOF_2D 1.09.01

Generated by Doxygen 1.8.13
on Wed Jul 8 2020 17:09:45

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

io	This package provides the classes needed to build the static parts of the user interface	19
model	This package provides the necessary classes to build a model for FullSWOF_UI parameters	19
model.definition	Each class in this package provides a single static method to instantiate a model tree corresponding to a FullSWOF configuration	21
parser	This package contains the parser and lexer used to parse mathematical formulas	22
ui	This package provides the controllers and views associated with the model classes	22
visualization	This package provides classes used to visualize FullSWOF output files	23
visualization.fs1d	This package provides the classes used to visualize FullSWOF_1D output files	25
visualization.fs2d	This package provides the classes used to visualize FullSWOF_2D output files	25

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

visualization.Animation< ChartContent >	34
visualization.AnimatedChart	27
visualization.Animation< SceneGraphComponent >	34
visualization.AnimatedScene	32
visualization.AnimationContent< E >	41
visualization.AnimationContent< ChartContent >	41
visualization.AnimatedChartContent	29
visualization.AnimationContent< SceneGraphComponent >	41
visualization.AnimatedSceneContent	34
visualization.ChartContent	70
visualization.ChartData	74
visualization.ChartLine	79
visualization.ChartScatterData	81
Comparable	
visualization.fs1d.OutputPoint1D	224
visualization.fs2d.OutputPoint2D	226
visualization.DataFileReader	83
model.definition.Definition_1D	85
model.definition.Definition_2D	86
DFA	
parser.FormulaLexer.DFA12	90
parser.FormulaLexer.DFA8	91
visualization.fs2d.FS2DFile.Format	126
io.FullswfIO	178
visualization.GnuplotFileReader< OutputPoint1D, TimeStep1D, TimeLine1D >	179
visualization.GnuplotFileReader< OutputPoint2D, TimeStep2D, TimeLine2D >	179
visualization.InputFileVisualizer	191
visualization.fs1d.HU1DVisualizer	188
visualization.fs1d.RainFileVisualizer	263
visualization.fs1d.Topography1DVisualizer	292
visualization.fs2d.HUV2DVisualizer	190
visualization.fs2d.RainFileVisualizer	263
visualization.fs2d.Topography2DVisualizer	294
model.Interval	201
visualization.JRealityViewingComponent	204

Lexer	
parser.FormulaLexer	133
model.ListFile	204
model.Node	213
model.ExternalNode	98
model.FieldParameter	106
model.DirectoryExtensionParameter	93
model.FileParameter	117
model.BoundaryParameter	56
model.FloatParameter	124
model.IntegerParameter	192
model.FileBuilderParameter	111
model.BoundaryFileParameter	46
model.FormulaFileBuilderParameter	129
model.HU1DBuilderParameter	187
model.HUV2DBuilderParameter	189
model.Topography1DBuilderParameter	291
model.Topography2DBuilderParameter	293
model.PointFileParameter	235
model.RainFileParameter	258
model.MultipleChoiceParameter	208
model.InternalNode	195
model.RootNode	269
visualization.OutputFileReader	221
visualization.fs2d.VtkFileReader2D	305
visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >	179
visualization.fs1d.GnuplotFileReader1D	182
visualization.fs2d.GnuplotFileReader2D	184
visualization.OutputPoint	223
visualization.fs1d.OutputPoint1D	224
visualization.fs2d.OutputPoint2D	226
Parser	
parser.FormulaParser	149
model.MultipleChoiceParameter.PossibleValue	242
io.Procedures	245
io.Start	275
visualization.TimeLine< TimeStep1D >	276
visualization.TimeLine< TimeStep2D >	276
visualization.TimeLine<?>	276
visualization.TimeStep< OutputPoint1D >	282
visualization.TimeStep< OutputPoint2D >	282
visualization.VisualizationFile< TimeLine1D >	295
visualization.VisualizationFile< TimeLine2D >	295
visualization.VisualizationFile<?>	295
ActionListener	
io.DataSetBuilderDialog	83
io.MainFrame	206
io.PreferencesDialog	244
io.ProgressDialog	255
ui.FileParameterController	121
ui.BoundaryFileParameterController	50

ui.FormulaFileBuilderController	127
ui.MultipleChoiceParameterController	212
visualization.Animation< E >	34
visualization.Chart	61
visualization.fs1d.FS1DVisualizationPane	174
visualization.fs2d.FS2DVisualizationPane	176
ChangeListener	
io.MainFrame	206
model.Dependency	87
model.DisablingDependency	95
model.EnablingDependency	96
model.SettingDependency	271
model.SettingDependency2	273
ui.ExternalNodeController	105
ui.FieldParameterController	108
ui.DirectoryExtensionController	92
ui.FileParameterController	121
ui.FileBuilderController	110
ui.BoundaryFileController	43
ui.PointFileController	233
ui.RainFileController	256
ui.FileBuilderWithoutSavedController	116
ui.FormulaFileBuilderController	127
ui.MultipleChoiceParameterController	212
visualization.VisualizationFile< E extends TimeLine<?>	295
visualization.fs1d.FS1DFile	173
visualization.fs2d.FS2DFile	175
Component	
visualization.Chart	61
DefaultTableCellRenderer	
ui.BoundaryFileController.CellRenderer	60
ui.PointFileController.CellRenderer	59
ui.RainFileController.CellRenderer	58
DefaultTableModel	
ui.BoundaryFileController.BoundaryModel	52
ui.PointFileController.PointModel	239
ui.RainFileController.RainModel	264
FocusListener	
ui.NodeController	217
ui.ExternalNodeController	105
ui.InternalNodeController	199
ui.ParametersGroupController	231
ui.RootController	268
HierarchyListener	
visualization.Animation< E >	34
JDialog	
io.DataSetBuilderDialog	83
io.HtmlAbout	186
io.PreferencesDialog	244
io.ProgressDialog	255
JFrame	
io.HtmlFrame	186
io.MainFrame	206

io.VisualizationFrame	299
JPanel	
visualization.Animation< E >	34
JTabbedPane	
visualization.VisualizationPane	301
visualization.fs1d.FS1DVisualizationPane	174
visualization.fs2d.FS2DVisualizationPane	176
LinkedList	
visualization.TimeLine< E extends TimeStep<?> >	276
visualization.fs1d.TimeLine1D	276
visualization.fs2d.TimeLine2D	279
MouseListener	
ui.BoundaryFileController	43
visualization.Chart	61
TreeSet	
visualization.TimeStep< E extends OutputPoint >	282
visualization.fs1d.TimeStep1D	284
visualization.fs2d.TimeStep2D	287
WindowListener	
io.MainFrame	206
io.VisualizationFrame	299

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

visualization.AnimatedChart	
An animated chart	27
visualization.AnimatedChartContent	
The content of an animated chart	29
visualization.AnimatedScene	
A three-dimensional animated chart	32
visualization.AnimatedSceneContent	
The content of a three dimensional animated chart	34
visualization.Animation< E >	
An abstract class to implement animations as Swing component	34
visualization.AnimationContent< E >	
The content of an animation	41
ui.BoundaryFileController	
A controller for a boundary file builder node	43
model.BoundaryFileParameter	
A parameter used to build a boundary file	46
ui.BoundaryFileParameterController	
A controller for a boundary file parameter node	50
ui.BoundaryFileController.BoundaryModel	
The model used by the view table	52
model.BoundaryParameter	
A parameter used to store a boundary file pathname	56
ui.RainFileController.CellRenderer	
An instance of this class is used to render the cells of the table in the view	58
ui.PointFileController.CellRenderer	
An instance of this class is used to render the cells of the table in the view	59
ui.BoundaryFileController.CellRenderer	
An instance of this class is used to render the cells of the table in the view	60
visualization.Chart	
A chart that can be displayed as a AWT component	61
visualization.ChartContent	
The content of a chart, that can include different types of data	70
visualization.ChartData	
An element of data to be displayed in a chart	74
visualization.ChartLine	
A line to be displayed on a chart	79

visualization.ChartScatterData	
A collection of points to be displayed in a scatter chart	81
visualization.DataFileReader	
Provides static method to read a simple data file and build a chart element from it	83
io.DataSetBuilderDialog	
A dialog box that allows the user to add custom data to a chart from a file	83
model.definition.Definition_1D	
This class provides a static method for generating the configuration used by FullSWOF_1D parameters files	85
model.definition.Definition_2D	
This class provides a static method for generating the configuration used by FullSWOF_2D parameters files	86
model.Dependency	
A dependency is a binary relationship between external nodes	87
parser.FormulaLexer.DFA12	90
parser.FormulaLexer.DFA8	91
ui.DirectoryExtensionController	
A controller for a directory extension parameter	92
model.DirectoryExtensionParameter	
A directory extension parameter is a special implementation of ExternalNode , which is typically used only once in a configuration tree	93
model.DisablingDependency	
A disabling dependency is used to disable the slave node when the master node is set to the target value	95
model.EnablingDependency	
An enabling dependency is used to enable the slave node when the master node is set to the target value	96
model.ExternalNode	
An external node in the tree model, typically a FullSWOF parameter	98
ui.ExternalNodeController	
A controller for an external node	105
model.FieldParameter	
This class provides the most permissive implementation of an external node, as any value will be considered a valid entry	106
ui.FieldParameterController	
A controller for a field parameter	108
ui.FileBuilderController	
A controller for a file builder parameter	110
model.FileBuilderParameter	
A parameter used to create an annex file	111
ui.FileBuilderWithoutSavedController	
A controller for a file builder parameter	116
model.FileParameter	
A parameter used to store a file pathname	117
ui.FileParameterController	
A controller for a file parameter node	121
model.FloatParameter	
A parameter with a floating point number value	124
visualization.fs2d.FS2DFile.Format	
The different formats of files produced by FullSWOF_2D	126
ui.FormulaFileBuilderController	
A controller for a file builder using parsed formulas	127

model.FormulaFileBuilderParameter	
A file builder parameter that uses a set of mathematical formulas to build a file	129
parser.FormulaLexer	
Transforms the character stream into a series of tokens	133
parser.FormulaParser	
Walks though the tokens to form mathematical sentences in the grammar	149
visualization.fs1d.FS1DFile	
A FullSWOF_1D output file	173
visualization.fs1d.FS1DVisualizationPane	
A visualization pane for a FullSWOF_1D output file	174
visualization.fs2d.FS2DFile	
A FullSWOF_2D output file	175
visualization.fs2d.FS2DVisualizationPane	
A tabbed pane presenting a FullSWOF_2D File	176
io.FullswofIO	
This class provides static methods to handle the interactions with the C++ FullSWOF code	178
visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >	
A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)	179
visualization.fs1d.GnuplotFileReader1D	
A reader for Gnuplot output files generated by FullSWOF_1D	182
visualization.fs2d.GnuplotFileReader2D	
A reader for Gnuplot output files generated by FullSWOF_2D	184
io.HtmlAbout	
An instance of this class is a JFrame used to display the content of About	186
io.HtmlFrame	
An instance of this class is a JFrame used to display the content of any HTML with basic style support	186
model.HU1DBuilderParameter	
A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u	187
visualization.fs1d.HU1DVisualizer	
A tool used to get a quick visualization of water input files for FullSWOF_1D	188
model.HUV2DBuilderParameter	
A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v	189
visualization.fs2d.HUV2DVisualizer	
A tool used to get a quick visualization of water input files for FullSWOF_2D	190
visualization.InputFileVisualizer	
A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)	191
model.IntegerParameter	
A parameter with an integer value	192
model.InternalNode	
This class can be used for any internal node of the tree	195
ui.InternalNodeController	
The controller of an internal node	199
model.Interval	
Describes a numerical interval	201
visualization.JRealityViewingComponent	
This class provides a static method to build a viewing component for a JReality SceneGraph↔ Component	204

model.ListFile	
The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : RainFileParameter , PointFileParametre , BoundaryFileparameter . . .	204
io.MainFrame	
An instance of this class is a JFrame corresponding to the main window of the user interface	206
model.MultipleChoiceParameter	
A parameter with a finite set of accepted values	208
ui.MultipleChoiceParameterController	
A controller for a multiple choice parameter node	212
model.Node	
A node in the model tree	213
ui.NodeController	
A controller for a node, in the model-view-controller pattern	217
visualization.OutputFileReader	
An abstract class to implement readers for FullSWOF output files	221
visualization.OutputPoint	
A cell in a FullSWOF output file	223
visualization.fs1d.OutputPoint1D	
A cell in a FullSWOF_1D output file	224
visualization.fs2d.OutputPoint2D	
A cell in a FullSWOF_2D output file	226
ui.ParametersGroupController	
A controller for an internal node	231
ui.PointFileController	
A controller for a point file builder node	233
model.PointFileParameter	
A parameter used to build a point file	235
ui.PointFileController.PointModel	
The model used by the view table	239
model.MultipleChoiceParameter.PossibleValue	
A possible value is constituted of two strings	242
io.PreferencesDialog	
An instance of this class is a JDialog corresponding to the preferences window of the user interface	244
io.Procedures	
This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project	245
io.ProgressDialog	
A dialog box including a progress bar, a console display and a cancel button	255
ui.RainFileController	
A controller for a rain file builder node	256
model.RainFileParameter	
A parameter used to build a rain file	258
visualization.fs1d.RainFileVisualizer	
A tool used to get a quick visualization of rain input files for FullSWOF_1D	263
visualization.fs2d.RainFileVisualizer	
A tool used to get a quick visualization of rain input files for FullSWOF_2D	263
ui.RainFileController.RainModel	
The model used by the view table	264
ui.RootController	
A controller for an internal node, especially suited for the root of the tree	268

model.RootNode	
This class can be used for any internal node, but its controller is better suited to the root of the tree	269
model.SettingDependency	
A setting dependency is used to set the slave node to a particular value when the master node is set to the target value	271
model.SettingDependency2	
A setting dependency is used to set the slave node to a particular value when the master node is set to the target value	273
io.Start	
The executable class used to launch the application	275
visualization.TimeLine< E extends TimeStep<?>>	
A collection of time step in a FullSWOF evolution file	276
visualization.fs1d.TimeLine1D	
A collection of TimeStep1D ordered by ascending time	276
visualization.fs2d.TimeLine2D	
A collection of TimeStep2D ordered by ascending time	279
visualization.TimeStep< E extends OutputPoint >	
A time step in a FullSWOF output file	282
visualization.fs1d.TimeStep1D	
A time step in a FullSWOF_1D output file	284
visualization.fs2d.TimeStep2D	
A time step in a FullSWOF_2D output file	287
model.Topography1DBuilderParameter	
A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z	291
visualization.fs1d.Topography1DVisualizer	
A tool used to get a quick visualization of topography input files for FullSWOF_1D	292
model.Topography2DBuilderParameter	
A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of z	293
visualization.fs2d.Topography2DVisualizer	
A tool used to get a quick visualization of topography input for FullSWOF_2D	294
visualization.VisualizationFile< E extends TimeLine<?>>	
.	295
io.VisualizationFrame	
An instance of this class is a JFrame used to visualize a FullSWOF output file	299
visualization.VisualizationPane	
A tabbed pane presenting a VisualizationFile	301
visualization.fs2d.VtkFileReader2D	
A reader for VTK output files generated by FullSWOF_2D	305

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/io/ DataSetBuilderDialog.java	
Dialog box	307
src/io/ FullswofIO.java	
Handle interactions with FullSWOF	307
src/io/ HtmlAbout.java	
Displays the content of About	308
src/io/ HtmlFrame.java	
How to display the content of an html file	309
src/io/ MainFrame.java	
Main window	310
src/io/ package-info.java	
Package for the static parts of the interface	310
src/io/ PreferencesDialog.java	
Preferences window	316
src/io/ Procedures.java	
Static methods	317
src/io/ ProgressDialog.java	
Dialog box with progress bar	318
src/io/ Start.java	
Start	319
src/io/ VisualizationFrame.java	
Visualization	319
src/model/ BoundaryFileParameter.java	
Builds a boundary file	320
src/model/ BoundaryParameter.java	
Stores a boundary file pathname	321
src/model/ Dependency.java	
Dependencies implementation	323
src/model/ DirectoryExtensionParameter.java	
Directory extension name used in FullSWOF	324
src/model/ DisablingDependency.java	
Disables dependencies	325
src/model/ EnablingDependency.java	
Enables dependencies	325
src/model/ ExternalNode.java	
External node	326

src/model/ FieldParameter.java	
Implements FullSWOF parameters	327
src/model/ FileBuilderParameter.java	
Implements parameters that are not in FullSWOF	327
src/model/ FileParameter.java	
Parameter that stores a file pathname	328
src/model/ FloatParameter.java	
Floating point parameter	329
src/model/ FormulaFileBuilderParameter.java	
Parameter with mathematical formulas	330
src/model/ HU1DBuilderParameter.java	
Writes hu files for FullSWOF_1D	331
src/model/ HUV2DBuilderParameter.java	
Writes huv files for FullSWOF_2D	331
src/model/ IntegerParameter.java	
Integer parameter	332
src/model/ InternalNode.java	
Internal node	333
src/model/ Interval.java	
Numerical interval	333
src/model/ ListFile.java	
For tables	334
src/model/ MultipleChoiceParameter.java	
Choice parameter (list of possible values)	335
src/model/ Node.java	
Node	336
src/model/ package-info.java	
Package for the parameters	312
src/model/ PointFileParameter.java	
Builds a point file	336
src/model/ RainFileParameter.java	
Parameter for rain file	337
src/model/ RootNode.java	
Node (preferentially root node)	338
src/model/ SettingDependency.java	
Sets the value of the slave node	339
src/model/ SettingDependency2.java	
Sets the value of the slave node taking into account error values	339
src/model/ Topography1DBuilderParameter.java	
Writes a FullSWOF_1D topography file from a formula	340
src/model/ Topography2DBuilderParameter.java	
Writes a FullSWOF_2D topography file from a formula	341
src/model/definition/ Definition_1D.java	
FullSWOF_1D configuration	322
src/model/definition/ Definition_2D.java	
FullSWOF_2D configuration	322
src/model/definition/ package-info.java	
Package to define FullSWOF configurations	311
src/parser/ FormulaLexer.java	
Lexer grammar	342
src/parser/ FormulaParser.java	
Parser	342

src/parser/ package-info.java	
Package for parsing the mathematical formulas	313
src/ui/ BoundaryFileController.java	
Controller for a boundary file builder node	343
src/ui/ BoundaryFileParameterController.java	
Controller for a boundary file parameter node	344
src/ui/ DirectoryExtensionController.java	
Controller for the directory extension name used in FullSWOF	344
src/ui/ ExternalNodeController.java	
External node controller	345
src/ui/ FieldParameterController.java	
Controller of FullSWOF parameters	346
src/ui/ FileBuilderController.java	
Controller for parameters that are not in FullSWOF	347
src/ui/ FileBuilderWithoutSavedController.java	
Controller for parameters that are not in FullSWOF, verifying errors	347
src/ui/ FileParameterController.java	
Controller for the parameter that stores a file pathname	348
src/ui/ FormulaFileBuilderController.java	
Controller for the parameter with mathematical formulas	349
src/ui/ InternalNodeController.java	
Controller of internal node	349
src/ui/ MultipleChoiceParameterController.java	
Controller for a choice parameter (list of possible values)	350
src/ui/ NodeController.java	
Controller of node	351
src/ui/ package-info.java	
Package for the controllers	314
src/ui/ ParametersGroupController.java	
Controller for parameters	352
src/ui/ PointFileController.java	
Controller for the parameter for point file builder node	352
src/ui/ RainFileController.java	
Controller for the parameter for rain file	353
src/ui/ RootController.java	
Controller for a node (preferentially root node)	354
src/visualization/ AnimatedChart.java	
Animated chart	355
src/visualization/ AnimatedChartContent.java	
Content of an animated chart	355
src/visualization/ AnimatedScene.java	
3D animated chart	356
src/visualization/ AnimatedSceneContent.java	
Content of a scene	357
src/visualization/ Animation.java	
Animation	357
src/visualization/ AnimationContent.java	
Content of animation	358
src/visualization/ Chart.java	
Chart	359
src/visualization/ ChartContent.java	
Content of chart	360

src/visualization/ ChartData.java	
Data for chart	360
src/visualization/ ChartLine.java	
Line in a chart	361
src/visualization/ ChartScatterData.java	
Data for scatter chart	362
src/visualization/ DataFileReader.java	
Reads data and builds chart	363
src/visualization/ GnuplotFileReader.java	
Gnuplot reader	376
src/visualization/ InputFileVisualizer.java	
Visualization of input files	377
src/visualization/ JRealityViewingComponent.java	
Builds a viewing component	378
src/visualization/ OutputFileReader.java	
Output reader	379
src/visualization/ OutputPoint.java	
Output point	379
src/visualization/ package-info.java	
Package for the visualization	316
src/visualization/ TimeLine.java	
Collection of time steps	380
src/visualization/ TimeStep.java	
Time step	381
src/visualization/ VisualizationFile.java	
Output file	382
src/visualization/ VisualizationPane.java	
Pane for the visualization	382
src/visualization/fs1d/ FS1DFile.java	
FullSWOF_1D file	363
src/visualization/fs1d/ FS1DVisualizationPane.java	
FullSWOF_1D visualization pane	364
src/visualization/fs1d/ GnuplotFileReader1D.java	
Gnuplot reader (1D files)	365
src/visualization/fs1d/ HU1DVisualizer.java	
Visualization of hu files	365
src/visualization/fs1d/ OutputPoint1D.java	
FullSWOF_1D cell	366
src/visualization/fs1d/ package-info.java	
Package for FullSWOF_1D visualization	315
src/visualization/fs1d/ RainFileVisualizer.java	
Visualization of rain (1D)	367
src/visualization/fs1d/ TimeLine1D.java	
Collection of 1D times steps	368
src/visualization/fs1d/ TimeStep1D.java	
Time step (1D)	369
src/visualization/fs1d/ Topography1DVisualizer.java	
Visualization of the topography (1D)	369
src/visualization/fs2d/ FS2DFile.java	
FullSWOF_2D file	370
src/visualization/fs2d/ FS2DVisualizationPane.java	
FullSWOF_2D visualization pane	371

src/visualization/fs2d/ GnuplotFileReader2D.java	
Gnuplot reader (2D files)	371
src/visualization/fs2d/ HUV2DVisualizer.java	
Visualization of huv files	372
src/visualization/fs2d/ OutputPoint2D.java	
FullSWOF_2D cell	373
src/visualization/fs2d/ package-info.java	
Package for FullSWOF_2D visualization	315
src/visualization/fs2d/ RainFileVisualizer.java	
Visualization of rain (2D)	367
src/visualization/fs2d/ TimeLine2D.java	
Collection of 2D times steps	374
src/visualization/fs2d/ TimeStep2D.java	
Time step (2D)	374
src/visualization/fs2d/ Topography2DVisualizer.java	
Visualization of the topography (2D)	375
src/visualization/fs2d/ VtkFileReader2D.java	
VTK reader (2D files)	376

Chapter 5

Namespace Documentation

5.1 Package io

This package provides the classes needed to build the static parts of the user interface.

Classes

- class [DataSetBuilderDialog](#)
A dialog box that allows the user to add custom data to a chart from a file.
- class [FullswofIO](#)
This class provides static methods to handle the interactions with the C++ FullSWOF code.
- class [HtmlAbout](#)
An instance of this class is a JFrame used to display the content of About.
- class [HtmlFrame](#)
An instance of this class is a JFrame used to display the content of any HTML with basic style support.
- class [MainFrame](#)
An instance of this class is a JFrame corresponding to the main window of the user interface.
- class [PreferencesDialog](#)
An instance of this class is a JDialog corresponding to the preferences window of the user interface.
- class [Procedures](#)
This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project.
- class [ProgressDialog](#)
A dialog box including a progress bar, a console display and a cancel button.
- class [Start](#)
The executable class used to launch the application.
- class [VisualizationFrame](#)
An instance of this class is a JFrame used to visualize a FullSWOF output file.

5.1.1 Detailed Description

This package provides the classes needed to build the static parts of the user interface.

This includes classes to build windows and their menus using swing components, but also the procedures to open and save files, run FullSWOF and render its output.

5.2 Package model

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

Packages

- package [definition](#)

Each class in this package provides a single static method to instantiate a model tree corresponding to a Full↔SWOF configuration.

Classes

- class [BoundaryFileParameter](#)

A parameter used to build a boundary file.

- class [BoundaryParameter](#)

A parameter used to store a boundary file pathname.

- class [Dependency](#)

A dependency is a binary relationship between external nodes.

- class [DirectoryExtensionParameter](#)

A directory extension parameter is a special implementation of [ExternalNode](#), which is typically used only once in a configuration tree.

- class [DisablingDependency](#)

A disabling dependency is used to disable the slave node when the master node is set to the target value.

- class [EnablingDependency](#)

An enabling dependency is used to enable the slave node when the master node is set to the target value.

- class [ExternalNode](#)

An external node in the tree model, typically a FullSWOF parameter.

- class [FieldParameter](#)

This class provides the most permissive implementation of an external node, as any value will be considered a valid entry.

- class [FileBuilderParameter](#)

A parameter used to create an annex file.

- class [FileParameter](#)

A parameter used to store a file pathname.

- class [FloatParameter](#)

A parameter with a floating point number value.

- class [FormulaFileBuilderParameter](#)

A file builder parameter that uses a set of mathematical formulas to build a file.

- class [HU1DBuilderParameter](#)

A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u.

- class [HUV2DBuilderParameter](#)

A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v.

- class [IntegerParameter](#)

A parameter with an integer value.

- class [InternalNode](#)

This class can be used for any internal node of the tree.

- class [Interval](#)

Describes a numerical interval.

- class [ListFile](#)

The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : [RainFileParameter](#), [PointFileParametre](#) , [BoundaryFileparameter](#).

- class [MultipleChoiceParameter](#)

A parameter with a finite set of accepted values.

- class [Node](#)

A node in the model tree.

- class [PointFileParameter](#)

A parameter used to build a point file.

- class [RainFileParameter](#)

A parameter used to build a rain file.

- class [RootNode](#)

This class can be used for any internal node, but its controller is better suited to the root of the tree.

- class [SettingDependency](#)

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

- class [SettingDependency2](#)

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

- class [Topography1DBuilderParameter](#)

A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z.

- class [Topography2DBuilderParameter](#)

A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of z.

5.2.1 Detailed Description

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

A model is organized as a composite tree, where all the objects extend the abstract class [Node](#). There are two types of nodes : internal nodes can have child nodes while external nodes cannot. External nodes usually represent parameters used in the interface, while internal nodes are groups of parameters (or groups of groups, since there is not limit to the depth of the tree). This package offers several implementations of external nodes, such as numeric parameters, string parameters, multiple choice parameters and many others, but you may need to implement new classes for more unusual purposes.

Besides nodes, this package also provides a [model.Dependency](#) abstract class. A dependency is a binary relationship between two external nodes that allow an action to be triggered under certain conditions.

See also

[model.definition](#) to see examples of model trees

[ui](#) for the corresponding controllers and views

5.3 Package model.definition

Each class in this package provides a single static method to instantiate a model tree corresponding to a Full↔SWOF configuration.

Classes

- class [Definition_1D](#)

This class provides a static method for generating the configuration used by FullSWOF_1D parameters files.

- class [Definition_2D](#)

This class provides a static method for generating the configuration used by FullSWOF_2D parameters files.

5.3.1 Detailed Description

Each class in this package provides a single static method to instantiate a model tree corresponding to a Full↔SWOF configuration.

To create a new configuration, create a new class with a static method to instantiate a node. Take for example the classes already provided in this package. To make the new configuration available in the user interface, you only need to add it to the array availableConfigurations in [io.Procedures](#) .

FullSWOF_UI is entirely internationalized. Each configuration should follow this pattern and you should avoid hardcoding the parameters name and description in the class. Instead use a resourceBundle, for which you must provide at least one default file. Place this file in a directory used only for this configuration, preferably in the I10n/config directory. You can provide more than one localization for a configuration, but new languages will only be displayed if the languages are also available for the user interface. For example if the user interface is localized in English and French, and you provide English and German localizations for a configuration, only the English localization will ever be used.

See also

`java.util.ResourceBundle`
[model](#)

5.4 Package parser

This package contains the parser and lexer used to parse mathematical formulas.

Classes

- class [FormulaLexer](#)
Transforms the character stream into a series of tokens.
- class [FormulaParser](#)
Walks though the tokens to form mathematical sentences in the grammar.

5.4.1 Detailed Description

This package contains the parser and lexer used to parse mathematical formulas.

These classes were auto-generated by ANTLR v3.4. Some methods inherited from BaseRecognizer are overridden so that exceptions are thrown instead of the default recovery mechanism.

See also

the file Formula.g for the grammar definition

5.5 Package ui

This package provides the controllers and views associated with the model classes.

Classes

- class [BoundaryFileController](#)
A controller for a boundary file builder node.
- class [BoundaryFileParameterController](#)
A controller for a boundary file parameter node.
- class [DirectoryExtensionController](#)
A controller for a directory extension parameter.
- class [ExternalNodeController](#)

- A controller for an external node.*

 - class [FieldParameterController](#)

A controller for a field parameter.
 - class [FileBuilderController](#)

A controller for a file builder parameter.
 - class [FileBuilderWithoutSavedController](#)

A controller for a file builder parameter.
 - class [FileParameterController](#)

A controller for a file parameter node.
 - class [FormulaFileBuilderController](#)

A controller for a file builder using parsed formulas.
 - class [InternalNodeController](#)

The controller of an internal node.
 - class [MultipleChoiceParameterController](#)

A controller for a multiple choice parameter node.
 - class [NodeController](#)

A controller for a node, in the model-view-controller pattern.
 - class [ParametersGroupController](#)

A controller for an internal node.
 - class [PointFileController](#)

A controller for a point file builder node.
 - class [RainFileController](#)

A controller for a rain file builder node.
 - class [RootController](#)

A controller for an internal node, especially suited for the root of the tree.

5.5.1 Detailed Description

This package provides the controllers and views associated with the model classes.

Controllers and views are closely linked, so there are no classes for views. Instead, each Controller has a view attribute, which it has to instantiate.

The controllers maintain a hierarchy which is parallel to that of the model. Internal node controllers must therefore maintain a list of child controllers, similar to the list of child nodes of their node. Likewise, views are modeled as a tree of JComponents.

The view never updates the model itself, it must fire an event (usually a FocusEvent to indicate that the view has lost focus after the user has finished writing his input) to the controller that will update the model. On the other hand, the model must fire a ChangeEvent to notify the controller that will update the view. Note that these procedures only apply to external nodes and their controllers, since internal nodes are not modified after their instantiation.

See also

javax.swing.JComponent
 javax.swing.event
[model](#)

5.6 Package visualization

This package provides classes used to visualize FullSWOF output files.

Packages

- package [fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.
- package [fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

Classes

- class [AnimatedChart](#)
An animated chart.
- class [AnimatedChartContent](#)
The content of an animated chart.
- class [AnimatedScene](#)
A three-dimensional animated chart.
- class [AnimatedSceneContent](#)
The content of a three dimensional animated chart.
- class [Animation](#)
An abstract class to implement animations as Swing component.
- class [AnimationContent](#)
The content of an animation.
- class [Chart](#)
A chart that can be displayed as a AWT component.
- class [ChartContent](#)
The content of a chart, that can include different types of data.
- class [ChartData](#)
An element of data to be displayed in a chart.
- class [ChartLine](#)
A line to be displayed on a chart.
- class [ChartScatterData](#)
A collection of points to be displayed in a scatter chart.
- class [DataFileReader](#)
Provides static method to read a simple data file and build a chart element from it.
- class [GnuplotFileReader](#)
A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)
- interface [InputFileVisualizer](#)
A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)
- class [JRealityViewingComponent](#)
This class provides a static method to build a viewing component for a JReality SceneGraphComponent.
- class [OutputFileReader](#)
An abstract class to implement readers for FullSWOF output files.
- class [OutputPoint](#)
A cell in a FullSWOF output file.
- class [TimeLine](#)
A collection of time step in a FullSWOF evolution file.
- class [TimeStep](#)
A time step in a FullSWOF output file.
- class [VisualizationFile](#)
- class [VisualizationPane](#)
A tabbed pane presenting a [VisualizationFile](#).

5.6.1 Detailed Description

This package provides classes used to visualize FullSWOF output files.

5.7 Package visualization.fs1d

This package provides the classes used to visualize FullSWOF_1D output files.

Classes

- class [FS1DFile](#)
A FullSWOF_1D output file.
- class [FS1DVisualizationPane](#)
A visualization pane for a FullSWOF_1D output file.
- class [GnuplotFileReader1D](#)
A reader for Gnuplot output files generated by FullSWOF_1D.
- class [HU1DVisualizer](#)
A tool used to get a quick visualization of water input files for FullSWOF_1D.
- class [OutputPoint1D](#)
A cell in a FullSWOF_1D output file.
- class [RainFileVisualizer](#)
A tool used to get a quick visualization of rain input files for FullSWOF_1D.
- class [TimeLine1D](#)
A collection of [TimeStep1D](#) ordered by ascending time.
- class [TimeStep1D](#)
A time step in a FullSWOF_1D output file.
- class [Topography1DVisualizer](#)
A tool used to get a quick visualization of topography input files for FullSWOF_1D.

5.7.1 Detailed Description

This package provides the classes used to visualize FullSWOF_1D output files.
The only format currently supported is the Gnuplot file format.

5.8 Package visualization.fs2d

This package provides the classes used to visualize FullSWOF_2D output files.

Classes

- class [FS2DFile](#)
A FullSWOF_2D output file.
- class [FS2DVisualizationPane](#)
A tabbed pane presenting a FullSWOF_2D File.
- class [GnuplotFileReader2D](#)
A reader for Gnuplot output files generated by FullSWOF_2D.
- class [HUV2DVisualizer](#)
A tool used to get a quick visualization of water input files for FullSWOF_2D.
- class [OutputPoint2D](#)

- A cell in a FullSWOF_2D output file.*
- class [RainFileVisualizer](#)
A tool used to get a quick visualization of rain input files for FullSWOF_2D.
- class [TimeLine2D](#)
A collection of [TimeStep2D](#) ordered by ascending time.
- class [TimeStep2D](#)
A time step in a FullSWOF_2D output file.
- class [Topography2DVisualizer](#)
A tool used to get a quick visualization of topography input for FullSWOF_2D.
- class [VtkFileReader2D](#)
A reader for VTK output files generated by FullSWOF_2D.

5.8.1 Detailed Description

This package provides the classes used to visualize FullSWOF_2D output files. Gnuplot files and VTK files are supported.

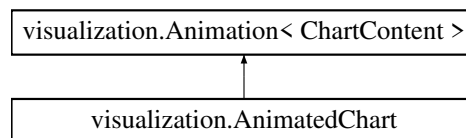
Chapter 6

Class Documentation

6.1 visualization.Animation< ChartContent > Class Reference

An animated chart.

Inheritance diagram for visualization.Animation:



Public Member Functions

- [Animation](#) ([AnimationContent](#) content)
- void [setContent](#) ([AnimationContent](#) content)
- void [setCurrentPosition](#) (int position)
- void [addImage](#) ([ChartContent](#) image)

Protected Member Functions

- Component [makeBackgroundComponent](#) ()

Additional Inherited Members

6.1.1 Detailed Description

An animated chart.

Definition at line 67 of file Animation.java.

6.1.2 Constructor & Destructor Documentation

Animation()

```
visualization.Animation.Animation (
    AnimationContent content )
```

Builds an animated chart with a content.

Parameters

<i>content</i>	the content of the chart
----------------	--------------------------

Definition at line 76 of file AnimatedChart.java.

6.1.3 Member Function Documentation**addImage()**

```
void visualization.AnimatedChart.addImage (
    ChartContent image )
```

Adds an image to the animation. This is a convenience method for `getContent().add(image)`.

Parameters

<i>image</i>	the image to be added
--------------	-----------------------

Definition at line 136 of file AnimatedChart.java.

makeBackgroundComponent()

```
Component visualization.AnimatedChart.makeBackgroundComponent ( ) [protected]
```

Builds the background of the animated chart.

Returns

an empty chart.

Definition at line 120 of file AnimatedChart.java.

setContent()

```
void visualization.AnimatedChart.setContent (
    AnimatedChartContent content )
```

Sets the content of the animated chart.

Parameters

<i>content</i>	the content of the animation
----------------	------------------------------

Definition at line 87 of file AnimatedChart.java.

setCurrentPosition()

```
void visualization.AnimatedChart.setCurrentPosition (
    int position )
```

Changes the image being shown.

Parameters

<i>position</i>	the new position. Must be a number between 0 and getImagesCount() or the method has no effect
-----------------	---

Definition at line 102 of file AnimatedChart.java.

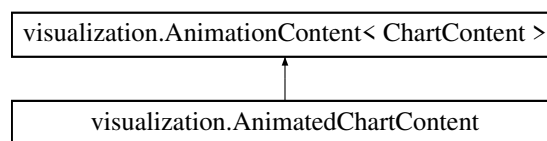
The documentation for this class was generated from the following file:

- [src/visualization/AnimatedChart.java](#)

6.2 visualization.AnimatedChartContent Class Reference

The content of an animated chart.

Inheritance diagram for visualization.AnimatedChartContent:

**Public Member Functions**

- [AnimatedChartContent](#) ()
- void [add](#) ([ChartContent](#) image)
- float [getXmax](#) ()
- float [getXmin](#) ()
- float [getYmax](#) ()
- float [getYmin](#) ()
- void [setBoundariesToInclude](#) (float x, float y)

6.2.1 Detailed Description

The content of an animated chart.

It is made of multiple chart content displayed one after the other in order to create an animation.

Definition at line 68 of file AnimatedChartContent.java.

6.2.2 Constructor & Destructor Documentation**AnimatedChartContent()**

```
visualization.AnimatedChartContent.AnimatedChartContent ( )
```

Constructs an empty animated chart content.

Definition at line 94 of file AnimatedChartContent.java.

6.2.3 Member Function Documentation

add()

```
void visualization.AnimatedChartContent.add (
    ChartContent image )
```

Adds an image at the end of the animation.

Parameters

<i>image</i>	the image to be added
--------------	-----------------------

Definition at line 104 of file AnimatedChartContent.java.

getXmax()

```
float visualization.AnimatedChartContent.getXmax ( )
```

Returns the greatest x coordinate of the data points among all chart contents.

Returns

the greatest x coordinate of the data points.

Definition at line 116 of file AnimatedChartContent.java.

getXmin()

```
float visualization.AnimatedChartContent.getXmin ( )
```

Returns the smallest x coordinate of the data points among all chart contents.

Returns

the smallest x coordinate of the data points.

Definition at line 126 of file AnimatedChartContent.java.

getYmax()

```
float visualization.AnimatedChartContent.getYmax ( )
```

Returns the greatest y coordinate of the data points among all chart contents.

Returns

the greatest y coordinate of the data points.

Definition at line 136 of file AnimatedChartContent.java.

getYmin()

```
float visualization.AnimatedChartContent.getYmin ( )
```

Returns the smallest y coordinate of the data points among all chart contents.

Returns

the smallest y coordinate of the data points.

Definition at line 146 of file AnimatedChartContent.java.

setBoundariesToInclude()

```
void visualization.AnimatedChartContent.setBoundariesToInclude (
    float x,
    float y )
```

Enlarges the boundaries of the chart if necessary, so that they include the coordinates.

Parameters

<i>x</i>	the x coordinate to include
<i>y</i>	the y coordinate to include

Definition at line 160 of file AnimatedChartContent.java.

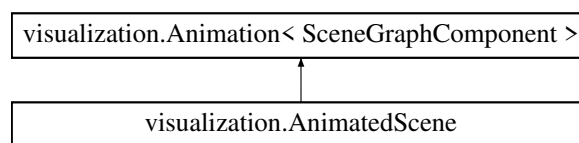
The documentation for this class was generated from the following file:

- src/visualization/[AnimatedChartContent.java](#)

6.3 visualization.Animation Class Reference

A three-dimensional animated chart.

Inheritance diagram for visualization.Animation:

**Public Member Functions**

- [AnimatedScene](#) ([AnimatedSceneContent](#) content)
- void [addImage](#) (SceneGraphComponent image)
- void [setContent](#) ([AnimationContent](#)< SceneGraphComponent > content)
- void [setCurrentPosition](#) (int position)

Protected Member Functions

- Component [makeBackgroundComponent](#) ()

Additional Inherited Members**6.3.1 Detailed Description**

A three-dimensional animated chart.

Definition at line 69 of file AnimatedScene.java.

6.3.2 Constructor & Destructor Documentation**AnimatedScene()**

```
visualization.Animation.Animation (
    AnimatedSceneContent content )
```

Builds an animated 3D graph with a content.

Parameters

<i>content</i>	the content of the animation
----------------	------------------------------

Definition at line 80 of file AnimatedScene.java.

6.3.3 Member Function Documentation

addImage()

```
void visualization.AnimatedScene.addImage (
    SceneGraphComponent image )
```

Adds an image at the end of the animation.

Parameters

<i>image</i>	the image to be added
--------------	-----------------------

Definition at line 92 of file AnimatedScene.java.

makeBackgroundComponent()

```
Component visualization.AnimatedScene.makeBackgroundComponent ( ) [protected]
```

Builds the viewer and returns it as a AWT component.

Returns

a viewing component.

Definition at line 142 of file AnimatedScene.java.

setContent()

```
void visualization.AnimatedScene.setContent (
    AnimationContent< SceneGraphComponent > content )
```

Sets the content of the animated chart.

Parameters

<i>content</i>	the content of the animation
----------------	------------------------------

Definition at line 105 of file AnimatedScene.java.

setCurrentPosition()

```
void visualization.AnimatedScene.setCurrentPosition (
    int position )
```

Changes the image being shown.

Parameters

<i>position</i>	the new position. Must be a number between 0 and getImagesCount() or the method has no effect
-----------------	---

Definition at line 122 of file AnimatedScene.java.

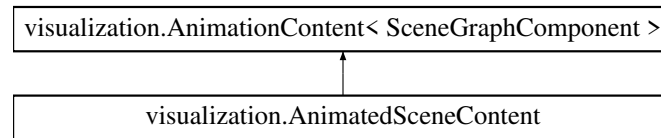
The documentation for this class was generated from the following file:

- src/visualization/[AnimatedScene.java](#)

6.4 visualization.AnimationSceneContent Class Reference

The content of a three dimensional animated chart.

Inheritance diagram for visualization.AnimationSceneContent:



Public Member Functions

- [AnimatedSceneContent](#) ()
- void [add](#) (SceneGraphComponent image)

6.4.1 Detailed Description

The content of a three dimensional animated chart.

This content is made of multiple 3D components, each made visible during a brief time to create an animation.

Definition at line 71 of file AnimatedSceneContent.java.

6.4.2 Constructor & Destructor Documentation

AnimatedSceneContent()

```
visualization.AnimationSceneContent.AnimationSceneContent ( )
```

Constructs an empty animated graph content.

Definition at line 77 of file AnimatedSceneContent.java.

6.4.3 Member Function Documentation

add()

```
void visualization.AnimationSceneContent.add (
    SceneGraphComponent image )
```

Adds an image to the end of the animation.

Parameters

<i>image</i>	the image to be added
--------------	-----------------------

Definition at line 88 of file AnimatedSceneContent.java.

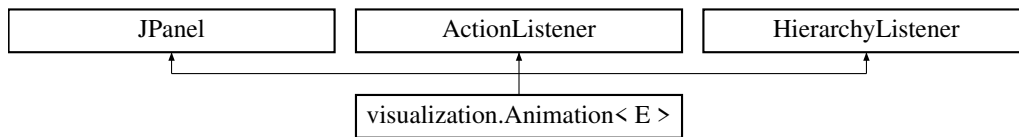
The documentation for this class was generated from the following file:

- src/visualization/[AnimatedSceneContent.java](#)

6.5 visualization.Animation< E > Class Template Reference

An abstract class to implement animations as Swing component.

Inheritance diagram for visualization.Animation< E >:



Public Member Functions

- void [actionPerformed](#) (ActionEvent evt)
- void [addImage](#) (E image)
- [AnimationContent](#)< E > [getContent](#) ()
- int [getCurrentPosition](#) ()
- int [getImagesCount](#) ()
- boolean [isPlaying](#) ()
- void [playOrPause](#) ()
- void [setContent](#) ([AnimationContent](#)< E > content)
- abstract void [setCurrentPosition](#) (int position)
- void [hierarchyChanged](#) (HierarchyEvent e)

Static Public Member Functions

- static int [getImagesPerSecond](#) ()
- static void [setImagesPerSecond](#) (int fps)

Protected Member Functions

- [Animation](#) ([AnimationContent](#)< E > content)
- JPanel [commandPanel](#) ()
- void [loadIcons](#) ()
- abstract Component [makeBackgroundComponent](#) ()

Protected Attributes

- JTextField [textField](#)
The text field that display which image is being shown.
- Component [backgroundComponent](#)
The animation background.
- [AnimationContent](#)< E > [content](#)
The content of this animation.
- int [currentPosition](#)
The number of the image being shown.
- Player [playerThread](#)
The thread used to change images.
- boolean [playing](#)
True if the animation is playing.
- JButton [playOrPauseButton](#)
The button used to launch the animation.
- ImageIcon [first](#)

Button icon.

- ImageIcon [previous](#)

Button icon.

- ImageIcon [play](#)

Button icon.

- ImageIcon [pause](#)

Button icon.

- ImageIcon [next](#)

Button icon.

- ImageIcon [last](#)

Button icon.

Static Protected Attributes

- static int [imagesPerSecond](#)

The rate of images per second of the animations.

6.5.1 Detailed Description

An abstract class to implement animations as Swing component.

The swing component includes a command panel to control the animation.

Implementation note: the animation must be stopped before the JPanel or its container are disposed of, otherwise the player thread may crash the application.

Parameters

<code><E></code>	the type of image used in the animation
------------------------	---

Definition at line 86 of file Animation.java.

6.5.2 Constructor & Destructor Documentation

Animation()

```
visualization.Animation< E >.Animation (
    AnimationContent< E > content ) [protected]
```

Builds an empty animation.

Definition at line 164 of file Animation.java.

6.5.3 Member Function Documentation

actionPerformed()

```
void visualization.Animation< E >.actionPerformed (
   (ActionEvent evt )
```

Called when the user interacts with the command panel.

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 183 of file Animation.java.

addImage()

```
void visualization.Animation< E >.addImage (
    E image )
```

Adds an image to the animation. This is a convenience method for `getContent().add(image)`.

Parameters

<i>image</i>	the image to be added
--------------	-----------------------

Definition at line 213 of file Animation.java.

commandPanel()

```
JPanel visualization.Animation< E >.commandPanel ( ) [protected]
```

Builds and returns the command panel where the user can interact.

Returns

a JPanel with control buttons.

Definition at line 313 of file Animation.java.

getContent()

```
AnimationContent<E> visualization.Animation< E >.getContent ( )
```

Returns

the content of the animation.

Definition at line 220 of file Animation.java.

getCurrentPosition()

```
int visualization.Animation< E >.getCurrentPosition ( )
```

Returns

the number of the image being shown.

Definition at line 227 of file Animation.java.

getImagesCount()

```
int visualization.Animation< E >.getImagesCount ( )
```

Returns the number of images in the animation. This is a convenience method for `getContent().getImagesCount()`.

Returns

the number of images in the animation.

Definition at line 238 of file Animation.java.

getImagesPerSecond()

```
static int visualization.Animation< E >.getImagesPerSecond ( ) [static]
```

Returns

the rate of images per second of the animations.

Definition at line 245 of file Animation.java.

hierarchyChanged()

```
void visualization.Animation< E >.hierarchyChanged (
    HierarchyEvent e )
```

Called when the component hierarchy is changed. If the JPanel is not displayable anymore (because its ancestor has been disposed) the animation MUST be stopped if it is still playing. Failure to do so might cause a threading problem.

Parameters

<i>e</i>	the triggering event
----------	----------------------

Definition at line 401 of file Animation.java.

isPlaying()

```
boolean visualization.Animation< E >.isPlaying ( )
```

Returns

true if the animation is currently playing.

Definition at line 252 of file Animation.java.

loadIcons()

```
void visualization.Animation< E >.loadIcons ( ) [protected]
```

Loads the button icon from image files.

Definition at line 346 of file Animation.java.

makeBackgroundComponent()

```
abstract Component visualization.Animation< E >.makeBackgroundComponent ( ) [abstract],
[protected]
```

Builds the background of the animation itself. This method is implementation-specific.

playOrPause()

```
void visualization.Animation< E >.playOrPause ( )
```

Starts the animation, or stops it if it is already playing.

Definition at line 260 of file Animation.java.

setContent()

```
void visualization.Animation< E >.setContent (
    AnimationContent< E > content )
```

Sets the content of the animation.

Parameters

<i>content</i>	the content of the animation
----------------	------------------------------

Definition at line 276 of file Animation.java.

setCurrentPosition()

```
abstract void visualization.Animation< E >.setCurrentPosition (
    int position ) [abstract]
```

Changes the image being shown. The way to actually change the image shown is implementation-specific so implementing classes must override this method.

Implementation note: this method should also change the value displayed in the command panel text field to reflect the change of image.

Parameters

<i>position</i>	the new position. Must be a number between 0 and getImagesCount() or the method has no effect
-----------------	---

setImagesPerSecond()

```
static void visualization.Animation< E >.setImagesPerSecond (
    int fps ) [static]
```

Sets the rate of images per second of the animations.

Parameters

<i>fps</i>	the rate of images per second
------------	-------------------------------

Definition at line 303 of file Animation.java.

6.5.4 Member Data Documentation

backgroundComponent

Component `visualization.Animation< E >.backgroundComponent` [protected]
The animation background.
Definition at line 98 of file Animation.java.

content

`AnimationContent<E> visualization.Animation< E >.content` [protected]
The content of this animation.
Definition at line 103 of file Animation.java.

currentPosition

int `visualization.Animation< E >.currentPosition` [protected]
The number of the image being shown.
Definition at line 113 of file Animation.java.

first

ImageIcon `visualization.Animation< E >.first` [protected]
Button icon.
Definition at line 133 of file Animation.java.

imagesPerSecond

int `visualization.Animation< E >.imagesPerSecond` [static], [protected]
The rate of images per second of the animations.
Definition at line 108 of file Animation.java.

last

ImageIcon `visualization.Animation< E >.last` [protected]
Button icon.
Definition at line 158 of file Animation.java.

next

ImageIcon `visualization.Animation< E >.next` [protected]
Button icon.
Definition at line 153 of file Animation.java.

pause

ImageIcon `visualization.Animation< E >.pause` [protected]
Button icon.
Definition at line 148 of file Animation.java.

play

ImageIcon [visualization.Animation< E >.play](#) [protected]

Button icon.

Definition at line 143 of file Animation.java.

playerThread

Player [visualization.Animation< E >.playerThread](#) [protected]

The thread used to change images.

Definition at line 118 of file Animation.java.

playing

boolean [visualization.Animation< E >.playing](#) [protected]

True if the animation is playing.

Definition at line 123 of file Animation.java.

playOrPauseButton

JButton [visualization.Animation< E >.playOrPauseButton](#) [protected]

The button used to launch the animation.

Definition at line 128 of file Animation.java.

previous

ImageIcon [visualization.Animation< E >.previous](#) [protected]

Button icon.

Definition at line 138 of file Animation.java.

textField

JTextField [visualization.Animation< E >.textField](#) [protected]

The text field that display which image is being shown.

It can also be used to jump to another image

Definition at line 93 of file Animation.java.

The documentation for this class was generated from the following file:

- [src/visualization/Animation.java](#)

6.6 visualization.AnimationContent< E > Class Template Reference

The content of an animation.

Public Member Functions

- [AnimationContent](#) ()
- void [add](#) (E image)
- E [get](#) (int i)
- int [getImagesCount](#) ()
- void [removeImages](#) ()

6.6.1 Detailed Description

The content of an animation.

Parameters

<code><E></code>	the type of images used by the animation
------------------------	--

Definition at line 70 of file AnimationContent.java.

6.6.2 Constructor & Destructor Documentation

AnimationContent()

```
visualization.AnimationContent< E >.AnimationContent ( )
```

Constructs an empty animation content.

Definition at line 83 of file AnimationContent.java.

6.6.3 Member Function Documentation

add()

```
void visualization.AnimationContent< E >.add (
    E image )
```

Adds an image at the end of the animation.

Parameters

<code>image</code>	the image to be added
--------------------	-----------------------

Definition at line 92 of file AnimationContent.java.

get()

```
E visualization.AnimationContent< E >.get (
    int i )
```

Returns the image at position i.

Parameters

<code>i</code>	the position of the image
----------------	---------------------------

Returns

an image (chart content).

Definition at line 104 of file AnimationContent.java.

getImagesCount()

```
int visualization.AnimationContent< E >.getImagesCount ( )
```

Returns

the number of images in this animation

Definition at line 111 of file AnimationContent.java.

removeImages()

```
void visualization.AnimationContent< E >.removeImages ( )
```

Removes all the images in the animation.

Definition at line 119 of file AnimationContent.java.

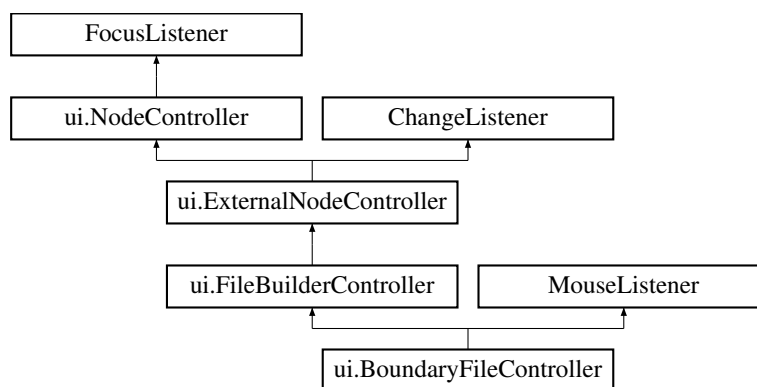
The documentation for this class was generated from the following file:

- src/visualization/[AnimationContent.java](#)

6.7 ui.BoundaryFileController Class Reference

A controller for a boundary file builder node.

Inheritance diagram for ui.BoundaryFileController:



Classes

- class [BoundaryModel](#)

The model used by the view table.

- class [CellRenderer](#)

An instance of this class is used to render the cells of the table in the view.

Public Member Functions

- [BoundaryFileController](#) ([BoundaryFileParameter](#) model)
- void [mouseClicked](#) (MouseEvent mouseEvent)
if you click about third column of the table, Open brown file and Write in the second column of the table
- void [mousePressed](#) (MouseEvent mouseEvent)
- void [mouseReleased](#) (MouseEvent mouseEvent)
- void [mouseEntered](#) (MouseEvent mouseEvent)
- void [mouseExited](#) (MouseEvent mouseEvent)
- boolean [validate](#) (File mainDirectory)

Package Functions

- void `highlightView` ()
- void `updateModel` ()
- void `updateView` ()
- void `setUpView` ()

Package Attributes

- JTable `viewTable`
The table of the view.
- JLabel `viewLabel`
The label of the view.

Additional Inherited Members

6.7.1 Detailed Description

A controller for a boundary file builder node.

This controller can set up a view that includes an editable table where the user can write time and file (or by brown file) value.

Definition at line 86 of file `BoundaryFileController.java`.

6.7.2 Constructor & Destructor Documentation

`BoundaryFileController()`

```
ui.BoundaryFileController.BoundaryFileController (
    BoundaryFileParameter model )
```

Constructs a controller for a `BoundaryFileParameter`.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 112 of file `BoundaryFileController.java`.

6.7.3 Member Function Documentation

`highlightView()`

```
void ui.BoundaryFileController.highlightView ( ) [package]
```

Puts the focus on the table.

Definition at line 121 of file `BoundaryFileController.java`.

`mouseClicked()`

```
void ui.BoundaryFileController.mouseClicked (
    MouseEvent mouseEvent )
```

if you click about third column of the table, Open brown file and Write in the second column of the table

Definition at line 166 of file BoundaryFileController.java.

mouseEntered()

```
void ui.BoundaryFileController.mouseEntered (
    MouseEvent mouseEvent )
```

Definition at line 193 of file BoundaryFileController.java.

mouseExited()

```
void ui.BoundaryFileController.mouseExited (
    MouseEvent mouseEvent )
```

Definition at line 198 of file BoundaryFileController.java.

mousePressed()

```
void ui.BoundaryFileController.mousePressed (
    MouseEvent mouseEvent )
```

Definition at line 183 of file BoundaryFileController.java.

mouseReleased()

```
void ui.BoundaryFileController.mouseReleased (
    MouseEvent mouseEvent )
```

Definition at line 188 of file BoundaryFileController.java.

setUpView()

```
void ui.BoundaryFileController.setUpView ( ) [package]
```

Definition at line 141 of file BoundaryFileController.java.

updateModel()

```
void ui.BoundaryFileController.updateModel ( ) [package]
```

Definition at line 126 of file BoundaryFileController.java.

updateView()

```
void ui.BoundaryFileController.updateView ( ) [package]
```

Definition at line 133 of file BoundaryFileController.java.

validate()

```
boolean ui.BoundaryFileController.validate (
    File mainDirectory )
```

Builds a file in the 'Inputs' directory located in mainDirectory according to the specifications of the file builder node. And copy files from boundary file in mainDirectory

Parameters

<i>mainDirectory</i>	the project directory located above the 'Inputs' directory
----------------------	--

Returns

true if the file has been correctly written.

Definition at line 551 of file BoundaryFileController.java.

6.7.4 Member Data Documentation**viewLabel**

`JLabel ui.BoundaryFileController.viewLabel [package]`

The label of the view.

Definition at line 103 of file BoundaryFileController.java.

viewTable

`JTable ui.BoundaryFileController.viewTable [package]`

The table of the view.

Definition at line 99 of file BoundaryFileController.java.

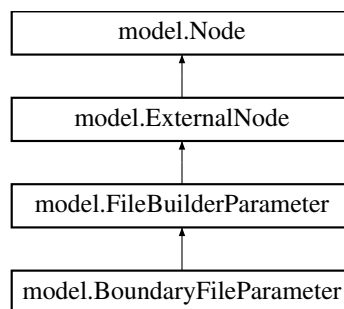
The documentation for this class was generated from the following file:

- [src/ui/BoundaryFileController.java](#)

6.8 model.BoundaryFileParameter Class Reference

A parameter used to build a boundary file.

Inheritance diagram for model.BoundaryFileParameter:

**Public Member Functions**

- [BoundaryFileParameter](#) (String [name](#), String [fileName](#), String boundaryTag, String boundaryFileTag, String boundaryTagValue)
Construct a boundary file builder parameter.
- boolean [fromFile](#) (File file) throws IOException
- void [readBoundaryFile](#) (File file)
- List< String > [getTime](#) ()

- List< String > [getFile](#) ()
- [NodeController](#) [setUpController](#) ()
- String [getFileContent](#) ()
- boolean [isValid](#) ()
- boolean [isValidTable](#) ()
- void [setTime](#) (List< String > time)
- void [setFile](#) (List< String > file)
- List< String > [getBrown](#) ()
- void [setBrown](#) (List< String > brown)
- String [getPath](#) ()

Static Public Member Functions

- static String [getTaggedValue](#) (String [tag](#), File file)

Additional Inherited Members

6.8.1 Detailed Description

A parameter used to build a boundary file.

Definition at line 73 of file BoundaryFileParameter.java.

6.8.2 Constructor & Destructor Documentation

BoundaryFileParameter()

```
model.BoundaryFileParameter.BoundaryFileParameter (
    String name,
    String fileName,
    String boundaryTag,
    String boundaryFileTag,
    String boundaryTagValue )
```

Construct a boundary file builder parameter.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file to be written
<i>boundaryTag</i>	the tag used for the boundary parameter
<i>boundaryFileTag</i>	the tag used for the boundary file parameter
<i>boundaryTagValue</i>	the value of the boundary parameter indicating that boundary is read from a file

Definition at line 125 of file BoundaryFileParameter.java.

6.8.3 Member Function Documentation

fromFile()

```
boolean model.BoundaryFileParameter.fromFile (
    File file ) throws IOException
```

If the parameters.txt file indicates that boundary is read from a file, this method will attempt to initialize this node values from that boundary file.

Definition at line 143 of file BoundaryFileParameter.java.

getBrown()

```
List<String> model.BoundaryFileParameter.getBrown ( )
```

Returns

the list of brown values to be written in the file.

Definition at line 354 of file BoundaryFileParameter.java.

getFile()

```
List<String> model.BoundaryFileParameter.getFile ( )
```

Returns

the list of file values to be written in the file.

Definition at line 234 of file BoundaryFileParameter.java.

getFileContent()

```
String model.BoundaryFileParameter.getFileContent ( )
```

Definition at line 244 of file BoundaryFileParameter.java.

getPath()

```
String model.BoundaryFileParameter.getPath ( )
```

Returns

absolute path

Definition at line 373 of file BoundaryFileParameter.java.

getTaggedValue()

```
static String model.BoundaryFileParameter.getTaggedValue (
    String tag,
    File file ) [static]
```

Returns the value associated with the tag in the specified parameters.txt file.

Parameters

<i>tag</i>	the tag to look for
<i>file</i>	the parameters.txt file to read

Returns

the value associated with the tag if it exists or an empty string otherwise.

Definition at line 201 of file BoundaryFileParameter.java.

getTime()

```
List<String> model.BoundaryFileParameter.getTime ( )
```

Returns

the list of time values to be written in the file.

Definition at line 227 of file BoundaryFileParameter.java.

isValid()

```
boolean model.BoundaryFileParameter.isValid ( )
```

Returns true if the list of values are valid. Lists must be the same length. The first value of the time list must be zero, and each value in this list greater than its predecessor. The file list must contain file name and each value exists. The brown list must contain only 'brown' The table must be Enabled and the last line must be empty

Returns

true if the list of values are valid.

Definition at line 269 of file BoundaryFileParameter.java.

isValidTable()

```
boolean model.BoundaryFileParameter.isValidTable ( )
```

Returns true if the list of values are valid. Lists must be the same length. The first value of the time list must be zero, and each value in this list greater than its predecessor. The file list must contain file name and each value exists. The brown list must contain only 'brown' The table can be enabled or disabled

Returns

true if the list of values are valid.

Definition at line 294 of file BoundaryFileParameter.java.

readBoundaryFile()

```
void model.BoundaryFileParameter.readBoundaryFile (
    File file )
```

Interprets a boundary file and initialize this node values accordingly.

Parameters

<i>file</i>	the boundary file to interpret
-------------	--------------------------------

Definition at line 160 of file BoundaryFileParameter.java.

setBrown()

```
void model.BoundaryFileParameter.setBrown (
    List< String > brown )
```

Sets the list of brown values to be written in the file.

Parameters

<i>brown</i>	the list of brown values to be written in the file
--------------	--

Definition at line 365 of file BoundaryFileParameter.java.

setFile()

```
void model.BoundaryFileParameter.setFile (
    List< String > file )
```

Sets the list of file values to be written in the file.

Parameters

<i>file</i>	the list of file values to be written in the file
-------------	---

Definition at line 346 of file BoundaryFileParameter.java.

setTime()

```
void model.BoundaryFileParameter.setTime (
    List< String > time )
```

Sets the list of time values to be written in the file.

Parameters

<i>time</i>	the list of time values to be written in the file
-------------	---

Definition at line 334 of file BoundaryFileParameter.java.

setUpController()

```
NodeController model.BoundaryFileParameter.setUpController ( )
```

Definition at line 239 of file BoundaryFileParameter.java.

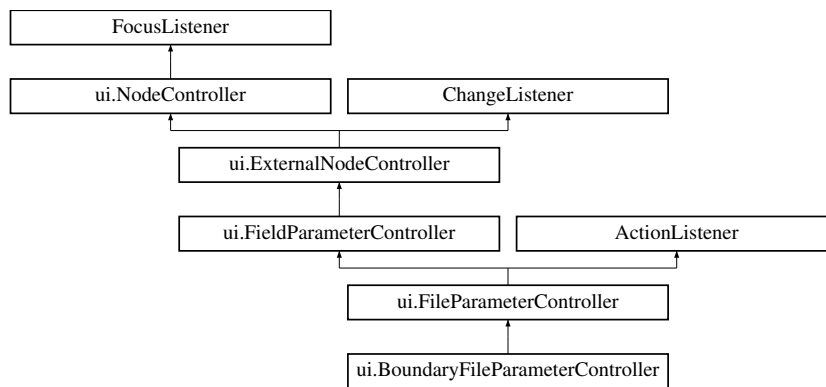
The documentation for this class was generated from the following file:

- [src/model/BoundaryFileParameter.java](#)

6.9 ui.BoundaryFileParameterController Class Reference

A controller for a boundary file parameter node.

Inheritance diagram for ui.BoundaryFileParameterController:



Public Member Functions

- [BoundaryFileParameterController](#) ([FileParameter](#) model)
- boolean [validate](#) (File mainDirectory)

Additional Inherited Members

6.9.1 Detailed Description

A controller for a boundary file parameter node.

This controller can set up a view suited for file browsing.

Definition at line 80 of file BoundaryFileParameterController.java.

6.9.2 Constructor & Destructor Documentation

BoundaryFileParameterController()

```
ui.BoundaryFileParameterController.BoundaryFileParameterController (
    FileParameter model )
```

Constructs a controller for a boundary file parameter node.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 89 of file BoundaryFileParameterController.java.

6.9.3 Member Function Documentation

validate()

```
boolean ui.BoundaryFileParameterController.validate (
    File mainDirectory )
```

Applies validation procedures to the node. This method is called when a project using this node is saved or run. The validation procedure for a file parameter includes copying the file designated by the model in mainDirectory if it is not already located there. Then the files in file parameter(boudouries) include copying or save in mainDirectory.

Parameters

<code>mainDirectory</code>	the directory in which to copy the file
----------------------------	---

Returns

true if files were successfully copied or was already in mainDirectory.

Exceptions

<code>IllegalStateException</code>	if mainDirectory is not a directory
------------------------------------	-------------------------------------

Definition at line 109 of file BoundaryFileParameterController.java.

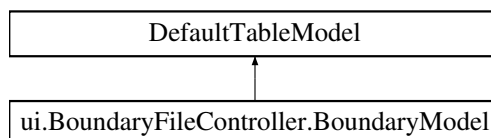
The documentation for this class was generated from the following file:

- [src/ui/BoundaryFileParameterController.java](#)

6.10 ui.BoundaryFileController.BoundaryModel Class Reference

The model used by the view table.

Inheritance diagram for ui.BoundaryFileController.BoundaryModel:

**Public Member Functions**

- [BoundaryModel](#) ()
- `Class<?>` [getColumnClass](#) (int columnIndex)
- int [getColumnCount](#) ()
- String [getColumnName](#) (int columnIndex)
- int [getRowCount](#) ()
- Object [getValueAt](#) (int rowIndex, int columnIndex)
- boolean [isCellEditable](#) (int rowIndex, int columnIndex)
- void [setValueAt](#) (Object aValue, int rowIndex, int columnIndex)
- boolean [isValidCell](#) (int row, int column)

Package Attributes

- List< String > [time](#)
The list of time values.
- List< String > [file](#)
The list of file values.

6.10.1 Detailed Description

The model used by the view table.

This model is an extension of the DefaultTableModel directly used by the JTable. It uses the model defined by BoundaryFileParameter to get its values.

Definition at line 289 of file BoundaryFileController.java.

6.10.2 Constructor & Destructor Documentation

BoundaryModel()

```
ui.BoundaryFileController.BoundaryModel.BoundaryModel ( )
```

Constructs a [BoundaryModel](#) for a JTable using the values stored in the BoundaryFileParameter.

Definition at line 308 of file BoundaryFileController.java.

6.10.3 Member Function Documentation

getColumnClass()

```
Class<?> ui.BoundaryFileController.BoundaryModel.getColumnClass (
    int columnIndex )
```

Returns the most specific superclass for all the cell values in the column, in this case String.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the common ancestor class of the object values in the model.

Definition at line 325 of file BoundaryFileController.java.

getColumnCount()

```
int ui.BoundaryFileController.BoundaryModel.getColumnCount ( )
```

Returns

the number of columns in the model.

Definition at line 333 of file BoundaryFileController.java.

getColumnName()

```
String ui.BoundaryFileController.BoundaryModel.getColumnName (
    int columnIndex )
```

Returns the name of the column at columnIndex.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the name of the column.

Definition at line 346 of file BoundaryFileController.java.

getRowCount()

```
int ui.BoundaryFileController.BoundaryModel.getRowCount ( )
```

Returns the number of rows in the model. That number includes an empty last line where the user can add new input.

Returns

the number of rows in the model.

Definition at line 368 of file BoundaryFileController.java.

getValueAt()

```
Object ui.BoundaryFileController.BoundaryModel.getValueAt (
    int rowIndex,
    int columnIndex )
```

Parameters

<i>rowIndex</i>	the row whose value is to be queried
<i>columnIndex</i>	the column whose value is to be queried

Returns

the value for the cell at columnIndex and rowIndex.

Definition at line 383 of file BoundaryFileController.java.

isCellEditable()

```
boolean ui.BoundaryFileController.BoundaryModel.isCellEditable (
    int rowIndex,
    int columnIndex )
```

Returns true if the cell at rowIndex and columnIndex is editable, which is the case except for cell (0,0)

Parameters

<i>rowIndex</i>	the row whose value to be queried
<i>columnIndex</i>	the column whose value to be queried

Returns

true if the cell at rowIndex and columnIndex is editable.

Definition at line 406 of file BoundaryFileController.java.

isValidCell()

```
boolean ui.BoundaryFileController.BoundaryModel.isValidCell (
    int row,
    int column )
```


Parameters

<i>row</i>	the row of the cell
<i>column</i>	the column of the cell

Returns

true if the value of the cell is valid.

```
if(f.exists()){ return true; }
    if(f.exists()){
    }
```

Definition at line 459 of file BoundaryFileController.java.

setValueAt()

```
void ui.BoundaryFileController.BoundaryModel.setValueAt (
    Object aValue,
    int rowIndex,
    int columnIndex )
```

Sets the value in the cell at columnIndex and rowIndex to aValue. This method also sets the value in the user model.

Parameters

<i>aValue</i>	the new value
<i>rowIndex</i>	the row whose value is to be changed
<i>columnIndex</i>	the column whose value is to be changed

Definition at line 424 of file BoundaryFileController.java.

6.10.4 Member Data Documentation**file**

```
List<String> ui.BoundaryFileController.BoundaryModel.file [package]
```

The list of file values.

Definition at line 297 of file BoundaryFileController.java.

time

```
List<String> ui.BoundaryFileController.BoundaryModel.time [package]
```

The list of time values.

Definition at line 293 of file BoundaryFileController.java.

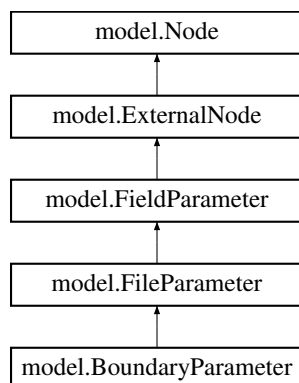
The documentation for this class was generated from the following file:

- [src/ui/BoundaryFileController.java](#)

6.11 model.BoundaryParameter Class Reference

A parameter used to store a boundary file pathname.

Inheritance diagram for model.BoundaryParameter:



Public Member Functions

- [BoundaryParameter](#) (String [name](#), String [tag](#), String [description](#))
- boolean [fromFile](#) (File file) throws IOException
- [NodeController](#) [setUpController](#) ()
- String [getPath](#) ()
- void [setPath](#) (String [path](#))

Package Attributes

- String [path](#)
the absolute pathname

Additional Inherited Members

6.11.1 Detailed Description

A parameter used to store a boundary file pathname.

The file must be declared by its absolute pathname and must exist at that location.

Definition at line 76 of file BoundaryParameter.java.

6.11.2 Constructor & Destructor Documentation

BoundaryParameter()

```

model.BoundaryParameter.BoundaryParameter (
    String name,
    String tag,
    String description )
  
```

Constructs a boundary file parameter with the provided name, tag and description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node

Definition at line 94 of file BoundaryParameter.java.

6.11.3 Member Function Documentation

fromFile()

```
boolean model.BoundaryParameter.fromFile (
    File file ) throws IOException
```

Attempts to set the values of the parameter from a file. Since files are only designated by their name in parameters.txt files, the location of that file is supposed to be the parent directory of the parameters.txt file.

Parameters

<i>file</i>	the parameters file containing the value
-------------	--

Returns

true if the parameter was successfully set.

Exceptions

<i>IOException</i>	if a problem occurred while reading the file, such as the file not being found
--------------------	--

Definition at line 111 of file BoundaryParameter.java.

getPath()

```
String model.BoundaryParameter.getPath ( )
```

Returns

the absolute pathname

Definition at line 151 of file BoundaryParameter.java.

setPath()

```
void model.BoundaryParameter.setPath (
    String path )
```

Parameters

<i>path</i>	the new absolute pathname
-------------	---------------------------

Definition at line 159 of file BoundaryParameter.java.

setUpController()

```
NodeController model.BoundaryParameter.setUpController ( )
```

Builds a controller for the node.

Returns

a node controller.

See also

[ui.BoundaryFileParameterController](#)

Definition at line 144 of file BoundaryParameter.java.

6.11.4 Member Data Documentation**path**

`String model.BoundaryParameter.path [package]`
the absolute pathname

Definition at line 81 of file BoundaryParameter.java.

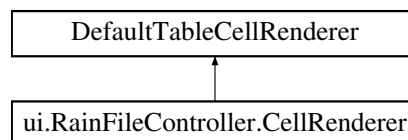
The documentation for this class was generated from the following file:

- `src/model/`[BoundaryParameter.java](#)

6.12 ui.RainFileController.CellRenderer Class Reference

An instance of this class is used to render the cells of the table in the view.

Inheritance diagram for `ui.RainFileController.CellRenderer`:

**Public Member Functions**

- Component [getTableCellRendererComponent](#) (JTable table, Object value, boolean selected, boolean focused, int row, int column)

6.12.1 Detailed Description

An instance of this class is used to render the cells of the table in the view.

See also

`javax.swing.table.DefaultTableCellRenderer`

Definition at line 172 of file RainFileController.java.

6.12.2 Member Function Documentation

getTableCellRendererComponent()

```
Component ui.RainFileController.CellRenderer.getTableCellRendererComponent (
    JTable table,
    Object value,
    boolean selected,
    boolean focused,
    int row,
    int column )
```

Returns the component used for drawing the cell. This method is used to configure the renderer appropriately before drawing.

Parameters

<i>table</i>	the JTable that is asking the renderer to draw; can be null
<i>value</i>	the value of the cell to be rendered. It is up to the specific renderer to interpret and draw the value. For example, if value is the string "true", it could be rendered as a string or it could be rendered as a check box that is checked. null is a valid value
<i>selected</i>	true if the cell is to be rendered with the selection highlighted; otherwise false
<i>focused</i>	if true, render cell appropriately. For example, put a special border on the cell, if the cell can be edited, render in the color used to indicate editing
<i>row</i>	the row index of the cell being drawn. When drawing the header, the value of row is -1
<i>column</i>	the column index of the cell being drawn

Definition at line 202 of file RainFileController.java.

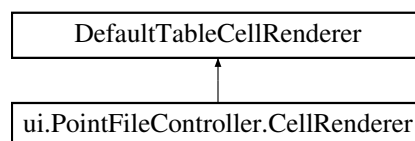
The documentation for this class was generated from the following file:

- [src/ui/RainFileController.java](#)

6.13 ui.PointFileController.CellRenderer Class Reference

An instance of this class is used to render the cells of the table in the view.

Inheritance diagram for ui.PointFileController.CellRenderer:

**Public Member Functions**

- Component [getTableCellRendererComponent](#) (JTable table, Object value, boolean selected, boolean focused, int row, int column)

6.13.1 Detailed Description

An instance of this class is used to render the cells of the table in the view.

See also

[javax.swing.table.DefaultTableCellRenderer](#)

Definition at line 162 of file PointFileController.java.

6.13.2 Member Function Documentation

getTableCellRendererComponent()

```
Component ui.PointFileController.CellRenderer.getTableCellRendererComponent (
    JTable table,
    Object value,
    boolean selected,
    boolean focused,
    int row,
    int column )
```

Returns the component used for drawing the cell. This method is used to configure the renderer appropriately before drawing.

Parameters

<i>table</i>	the JTable that is asking the renderer to draw; can be null
<i>value</i>	the value of the cell to be rendered. It is up to the specific renderer to interpret and draw the value. For example, if value is the string "true", it could be rendered as a string or it could be rendered as a check box that is checked. null is a valid value
<i>selected</i>	true if the cell is to be rendered with the selection highlighted; otherwise false
<i>focused</i>	if true, render cell appropriately. For example, put a special border on the cell, if the cell can be edited, render in the color used to indicate editing
<i>row</i>	the row index of the cell being drawn. When drawing the header, the value of row is -1
<i>column</i>	the column index of the cell being drawn

Definition at line 192 of file PointFileController.java.

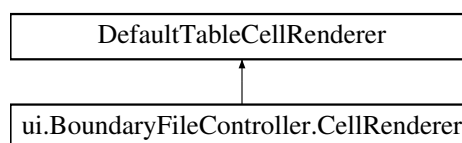
The documentation for this class was generated from the following file:

- src/ui/[PointFileController.java](#)

6.14 ui.BoundaryFileController.CellRenderer Class Reference

An instance of this class is used to render the cells of the table in the view.

Inheritance diagram for ui.BoundaryFileController.CellRenderer:



Public Member Functions

- Component [getTableCellRendererComponent](#) (JTable table, Object value, boolean selected, boolean focused, int row, int column)

6.14.1 Detailed Description

An instance of this class is used to render the cells of the table in the view.

See also

`javax.swing.table.DefaultTableCellRenderer`

Definition at line 210 of file `BoundaryFileController.java`.

6.14.2 Member Function Documentation

`getTableCellRendererComponent()`

```
Component ui.BoundaryFileController.CellRenderer.getTableCellRendererComponent (
    JTable table,
    Object value,
    boolean selected,
    boolean focused,
    int row,
    int column )
```

Returns the component used for drawing the cell. This method is used to configure the renderer appropriately before drawing.

Parameters

<i>table</i>	the JTable that is asking the renderer to draw; can be null
<i>value</i>	the value of the cell to be rendered. It is up to the specific renderer to interpret and draw the value. For example, if value is the string "true", it could be rendered as a string or it could be rendered as a check box that is checked. null is a valid value
<i>selected</i>	true if the cell is to be rendered with the selection highlighted; otherwise false
<i>focused</i>	if true, render cell appropriately. For example, put a special border on the cell, if the cell can be edited, render in the color used to indicate editing
<i>row</i>	the row index of the cell being drawn. When drawing the header, the value of row is -1
<i>column</i>	the column index of the cell being drawn

Definition at line 240 of file `BoundaryFileController.java`.

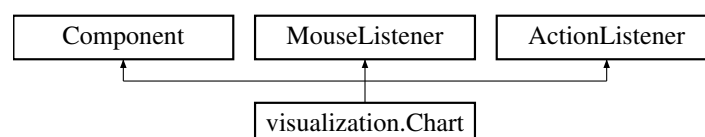
The documentation for this class was generated from the following file:

- [src/ui/BoundaryFileController.java](#)

6.15 visualization.Chart Class Reference

A chart that can be displayed as a AWT component.

Inheritance diagram for `visualization.Chart`:



Public Member Functions

- [Chart \(\)](#)
- [Chart \(ChartContent content\)](#)

- void [actionPerformed](#) (ActionEvent evt)
- void [add](#) (ChartData e)
- String [getTitle](#) ()
- void [setTitle](#) (String title)
- void [clear](#) ()
- [ChartContent](#) [getContent](#) ()
- BufferedImage [getImage](#) ()
- Dimension [getPreferredSize](#) ()
- Dimension [getSize](#) ()
- float [getXmax](#) ()
- float [getXmin](#) ()
- float [getYmax](#) ()
- float [getYmin](#) ()
- void [mouseClicked](#) (MouseEvent evt)
- void [mouseEntered](#) (MouseEvent evt)
- void [mouseExited](#) (MouseEvent evt)
- void [mousePressed](#) (MouseEvent evt)
- void [mouseReleased](#) (MouseEvent evt)
- void [paint](#) (Graphics g)
- void [saveImage](#) ()
- void [setBoundaries](#) ()
- void [setBoundaries](#) (float xmin, float xmax, float ymin, float ymax)
- void [setBoundariesToInclude](#) (float x, float y)
- void [setBounds](#) (int x, int y, int width, int height)
- void [setContent](#) ([ChartContent](#) content, boolean resize)
- void [setPreferredSize](#) (Dimension d)
- void [setSize](#) (Dimension d)
- void [setSize](#) (int width, int height)
- void [showPopupMenu](#) (MouseEvent evt)

Protected Member Functions

- Line2D.Float [scaledLine](#) (float x1, float y1, float x2, float y2)
- Point2D.Float [scaledPoint](#) (float x, float y)
- Point2D.Float [scaledPoint](#) (Point2D.Float p)

6.15.1 Detailed Description

A chart that can be displayed as a AWT component.

Definition at line 100 of file Chart.java.

6.15.2 Constructor & Destructor Documentation

Chart() [1/2]

```
visualization.Chart.Chart ( )
```

Constructs an empty chart.

Definition at line 174 of file Chart.java.

Chart() [2/2]

```
visualization.Chart.Chart (
    ChartContent content )
```

Constructs a chart with a content.

Parameters

<i>content</i>	the content of the chart
----------------	--------------------------

Definition at line 190 of file Chart.java.

6.15.3 Member Function Documentation**actionPerformed()**

```
void visualization.Chart.actionPerformed (
    ActionEvent evt )
```

Called when a popup menu item is clicked.

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 206 of file Chart.java.

add()

```
void visualization.Chart.add (
    ChartData e )
```

Adds an element to the chart. This is a convenience method for [getContent\(\).add\(e\)](#).

Parameters

<i>e</i>	the line to be added
----------	----------------------

Definition at line 232 of file Chart.java.

clear()

```
void visualization.Chart.clear ( )
```

Clears the content of the chart. This is a convenience method for [getContent\(\).clear\(\)](#).

Definition at line 259 of file Chart.java.

getContent()

```
ChartContent visualization.Chart.getContent ( )
```

Returns

the content of the chart.

Definition at line 266 of file Chart.java.

getImage()

```
BufferedImage visualization.Chart.getImage ( )
```

Returns a buffered image of the chart. The size of the image is the same as that of the component.

Returns

a buffered image of the chart.

Definition at line 277 of file Chart.java.

getPreferredSize()

```
Dimension visualization.Chart.getPreferredSize ( )
```

Returns the preferred size of the component, which is the same as its current size.

Returns

the preferred size.

Definition at line 293 of file Chart.java.

getSize()

```
Dimension visualization.Chart.getSize ( )
```

Returns

the size of the component.

Definition at line 301 of file Chart.java.

getTitle()

```
String visualization.Chart.getTitle ( )
```

Returns

the chart title.

Definition at line 239 of file Chart.java.

getXmax()

```
float visualization.Chart.getXmax ( )
```

Returns

the greatest x coordinate displayed in the plotting area.

Definition at line 308 of file Chart.java.

getXmin()

```
float visualization.Chart.getXmin ( )
```

Returns

the smallest x coordinate displayed in the plotting area.

Definition at line 315 of file Chart.java.

getYmax()

```
float visualization.Chart.getYmax ( )
```

Returns

the greatest y coordinate displayed in the plotting area.

Definition at line 322 of file Chart.java.

getYmin()

```
float visualization.Chart.getYmin ( )
```

Returns

the smallest y coordinate displayed in the plotting area.

Definition at line 329 of file Chart.java.

mouseClicked()

```
void visualization.Chart.mouseClicked (
    MouseEvent evt )
```

May trigger the popup menu (the trigger is platform-specific).

Definition at line 337 of file Chart.java.

mouseEntered()

```
void visualization.Chart.mouseEntered (
    MouseEvent evt )
```

May trigger the popup menu (the trigger is platform-specific).

Definition at line 347 of file Chart.java.

mouseExited()

```
void visualization.Chart.mouseExited (
    MouseEvent evt )
```

May trigger the popup menu (the trigger is platform-specific).

Definition at line 357 of file Chart.java.

mousePressed()

```
void visualization.Chart.mousePressed (
    MouseEvent evt )
```

May trigger the popup menu (the trigger is platform-specific).

Definition at line 367 of file Chart.java.

mouseReleased()

```
void visualization.Chart.mouseReleased (
    MouseEvent evt )
```

May trigger the popup menu (the trigger is platform-specific).

Definition at line 377 of file Chart.java.

paint()

```
void visualization.Chart.paint (
    Graphics g )
```

Paint the chart.

Parameters

<i>g</i>	the Graphics context in which to paint
----------	--

Definition at line 391 of file Chart.java.

saveImage()

```
void visualization.Chart.saveImage ( )
```

Allows the user to save the chart as an image file.

Definition at line 411 of file Chart.java.

scaledLine()

```
Line2D.Float visualization.Chart.scaledLine (
    float x1,
    float y1,
    float x2,
    float y2 ) [protected]
```

Returns a line with coordinates converted to drawing coordinates.

Parameters

<i>x1</i>	the original x coordinate of one end of the line
<i>y1</i>	the original y coordinate of the other end of the line
<i>x2</i>	the original x coordinate of one end of the line
<i>y2</i>	the original y coordinate of the other end of the line

Returns

a line with scaled coordinates.

Definition at line 634 of file Chart.java.

scaledPoint() [1/2]

```
Point2D.Float visualization.Chart.scaledPoint (
    float x,
    float y ) [protected]
```

Returns a point with coordinates converted to drawing coordinates.

Parameters

<i>x</i>	the original x coordinate of the point
<i>y</i>	the original y coordinate of the point

Returns

a point with scaled coordinates.

Definition at line 648 of file Chart.java.

scaledPoint() [2/2]

```
Point2D.Float visualization.Chart.scaledPoint (
    Point2D.Float p ) [protected]
```

Returns a point with coordinates converted to drawing coordinates.

Parameters

<i>p</i>	the original point
----------	--------------------

Returns

a point with scaled coordinates.

Definition at line 663 of file Chart.java.

setBoundaries() [1/2]

```
void visualization.Chart.setBoundaries ( )
```

Changes the boundaries of the plotting area so that all the content will fit in it.

Definition at line 452 of file Chart.java.

setBoundaries() [2/2]

```
void visualization.Chart.setBoundaries (
    float xmin,
    float xmax,
```

```
float ymin,
float ymax )
```

Changes the boundaries of the plotting area. Points with coordinates outside of these boundaries can be drawn outside of this area and might appear in the margins or outside of the component.

Parameters

<i>xmin</i>	the smallest x-coordinate to appear in the plotting area
<i>xmax</i>	the greatest x-coordinate to appear in the plotting area
<i>ymin</i>	the smallest y-coordinate to appear in the plotting area
<i>ymax</i>	the greatest y-coordinate to appear in the plotting area

Definition at line 472 of file Chart.java.

setBoundariesToInclude()

```
void visualization.Chart.setBoundariesToInclude (
    float x,
    float y )
```

Changes the boundaries of the plotting area to include a point. Boundaries can only be made larger by this method.

Parameters

<i>x</i>	the x-coordinate of the point to appear in the plotting area
<i>y</i>	the y-coordinate of the point to appear in the plotting area

Definition at line 504 of file Chart.java.

setBounds()

```
void visualization.Chart.setBounds (
    int x,
    int y,
    int width,
    int height )
```

Moves and resizes this component.

Parameters

<i>x</i>	the new x-coordinate of this component
<i>y</i>	the new y-coordinate of this component
<i>width</i>	the new width of this component
<i>height</i>	the new height of this component

Definition at line 544 of file Chart.java.

setContent()

```
void visualization.Chart.setContent (
    ChartContent content,
```

```
boolean resize )
```

Sets the content of the chart.

Parameters

<i>content</i>	the content of the chart
<i>resize</i>	if true the boundaries of the chart will be moved to display all of the new content

Definition at line 559 of file Chart.java.

setPreferredSize()

```
void visualization.Chart.setPreferredSize (
    Dimension d )
```

Sets the preferred size of the component.

Parameters

<i>d</i>	the new size of the component
----------	-------------------------------

Definition at line 575 of file Chart.java.

setSize() [1/2]

```
void visualization.Chart.setSize (
    Dimension d )
```

Sets the preferred size of the component.

Parameters

<i>d</i>	the new size of the component
----------	-------------------------------

Definition at line 587 of file Chart.java.

setSize() [2/2]

```
void visualization.Chart.setSize (
    int width,
    int height )
```

Sets the size of the component.

Parameters

<i>width</i>	the new width of the component
<i>height</i>	the new height of the component

Definition at line 601 of file Chart.java.

setTitle()

```
void visualization.Chart.setTitle (
```

```
String title )
```

Sets the chart title.

Parameters

<i>title</i>	the title of the chart
--------------	------------------------

Definition at line 250 of file Chart.java.

showPopupMenu()

```
void visualization.Chart.showPopupMenu (
    MouseEvent evt )
```

Shows a popup menu that allows the user to save the chart as an image file.

Parameters

<i>evt</i>	the triggering mouse event
------------	----------------------------

Definition at line 615 of file Chart.java.

The documentation for this class was generated from the following file:

- src/visualization/[Chart.java](#)

6.16 visualization.ChartContent Class Reference

The content of a chart, that can include different types of data.

Public Member Functions

- [ChartContent](#) ()
- [ChartContent](#) (String chartTitle)
- void [add](#) ([ChartData](#) d)
- void [clear](#) ()
- [Chart](#) [getContainer](#) ()
- void [remove](#) ([ChartData](#) d)
- List< [ChartData](#) > [getData](#) ()
- float [getXmax](#) ()
- float [getXmin](#) ()
- float [getYmax](#) ()
- float [getYmin](#) ()
- void [setContainer](#) ([Chart](#) container)
- void [setChartTitle](#) (String chartTitle)
- String [getChartTitle](#) ()

Protected Member Functions

- void [paint](#) (Graphics2D g)

6.16.1 Detailed Description

The content of a chart, that can include different types of data.

Definition at line 68 of file ChartContent.java.

6.16.2 Constructor & Destructor Documentation

ChartContent() [1/2]

```
visualization.ChartContent.ChartContent ( )
```

Constructs an empty chart content.

Definition at line 109 of file ChartContent.java.

ChartContent() [2/2]

```
visualization.ChartContent.ChartContent (
    String chartTitle )
```

Constructs an empty chart content with a title.

Parameters

<i>chartTitle</i>	the chart title
-------------------	-----------------

Definition at line 120 of file ChartContent.java.

6.16.3 Member Function Documentation

add()

```
void visualization.ChartContent.add (
    ChartData d )
```

Adds an element of data to the chart content.

Parameters

<i>d</i>	the element of data to be added
----------	---------------------------------

Definition at line 132 of file ChartContent.java.

clear()

```
void visualization.ChartContent.clear ( )
```

Clears all data in this chart content.

Definition at line 145 of file ChartContent.java.

getChartTitle()

```
String visualization.ChartContent.getChartTitle ( )
```

Returns

the chart title.

Definition at line 272 of file ChartContent.java.

getContainer()

```
Chart visualization.ChartContent.getContainer ( )
```

Returns the container chart, which can be null.

Returns

the container chart.

Definition at line 159 of file ChartContent.java.

getData()

```
List<ChartData> visualization.ChartContent.getData ( )
```

Returns

the list of data elements displayed on the chart.

Definition at line 179 of file ChartContent.java.

getXmax()

```
float visualization.ChartContent.getXmax ( )
```

Returns

the greatest x coordinate of the data points.

Definition at line 186 of file ChartContent.java.

getXmin()

```
float visualization.ChartContent.getXmin ( )
```

Returns

the smallest x coordinate of the data points.

Definition at line 193 of file ChartContent.java.

getYmax()

```
float visualization.ChartContent.getYmax ( )
```

Returns

the greatest y coordinate of the data points.

Definition at line 200 of file ChartContent.java.

getYmin()

```
float visualization.ChartContent.getYmin ( )
```

Returns

the smallest y coordinate of the data points.

Definition at line 207 of file ChartContent.java.

paint()

```
void visualization.ChartContent.paint (
    Graphics2D g ) [protected]
    Paints the date/
```

Parameters

<i>g</i>	the Graphics context in which to paint
----------	--

Definition at line 230 of file ChartContent.java.

remove()

```
void visualization.ChartContent.remove (
    ChartData d )
    Removes a chart element from the content.
```

Parameters

<i>d</i>	the element to remove
----------	-----------------------

Definition at line 170 of file ChartContent.java.

setChartTitle()

```
void visualization.ChartContent.setChartTitle (
    String chartTitle )
    Sets the chart title.
```

Parameters

<i>chartTitle</i>	the title of the chart
-------------------	------------------------

Definition at line 265 of file ChartContent.java.

setContainer()

```
void visualization.ChartContent.setContainer (
    Chart container )
    Sets the container chart. This method must be called before the data is painted.
```

Parameters

<i>container</i>	the container chart
------------------	---------------------

Definition at line 219 of file ChartContent.java.

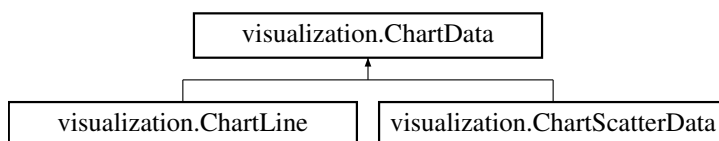
The documentation for this class was generated from the following file:

- src/visualization/[ChartContent.java](#)

6.17 visualization.ChartData Class Reference

An element of data to be displayed in a chart.

Inheritance diagram for visualization.ChartData:



Public Member Functions

- [ChartData](#) ()
- [ChartData](#) (String [name](#))
- [ChartData](#) (String [name](#), Color [color](#))
- void [add](#) (float x, float y)
- Color [getColor](#) ()
- [ChartContent](#) [getContainer](#) ()
- String [getName](#) ()
- float [getXmax](#) ()
- float [getXmin](#) ()
- float [getYmax](#) ()
- float [getYmin](#) ()
- abstract void [paint](#) (Graphics2D g)
- abstract void [paintLegendElement](#) (Graphics2D g, int x, int y)
- void [setBoundariesToInclude](#) (float x, float y)
- void [setColor](#) (Color [color](#))
- void [setContainer](#) ([ChartContent](#) [container](#))
- void [setName](#) (String [name](#))

Protected Attributes

- [ChartContent](#) [container](#)
The chart content to which the element belongs.
- String [name](#)
The name of the element.
- Color [color](#)
The color in which to paint the element.
- List< Point2D.Float > [data](#)
The list of points included in this element.

6.17.1 Detailed Description

An element of data to be displayed in a chart.

Definition at line 72 of file ChartData.java.

6.17.2 Constructor & Destructor Documentation

ChartData() [1/3]

```
visualization.ChartData.ChartData ( )
```

Constructs an empty data element.

Definition at line 119 of file ChartData.java.

ChartData() [2/3]

```
visualization.ChartData.ChartData (
    String name )
```

Constructs an empty data element with a name.

Parameters

<i>name</i>	the name of the element
-------------	-------------------------

Definition at line 130 of file ChartData.java.

ChartData() [3/3]

```
visualization.ChartData.ChartData (
    String name,
    Color color )
```

Constructs an empty data element with a name and a color.

Parameters

<i>name</i>	the name of the element
<i>color</i>	the color in which to paint the element

Definition at line 144 of file ChartData.java.

6.17.3 Member Function Documentation**add()**

```
void visualization.ChartData.add (
    float x,
    float y )
```

Adds a point to the data element.

Parameters

<i>x</i>	the x coordinate of the point
<i>y</i>	the y coordinate of the point

Definition at line 159 of file ChartData.java.

getColor()

```
Color visualization.ChartData.getColor ( )
```

Returns

the color in which the data element is painted.

Definition at line 179 of file ChartData.java.

getContainer()

```
ChartContent visualization.ChartData.getContainer ( )
```

Returns the chart content to which the element belongs. Can be null.

Returns

the chart content to which the element belongs.

Definition at line 189 of file ChartData.java.

getName()

```
String visualization.ChartData.getName ( )
```

Returns

the name of the element.

Definition at line 196 of file ChartData.java.

getXmax()

```
float visualization.ChartData.getXmax ( )
```

Returns

the greatest x coordinate of the points.

Definition at line 203 of file ChartData.java.

getXmin()

```
float visualization.ChartData.getXmin ( )
```

Returns

the smallest x coordinate of the points.

Definition at line 210 of file ChartData.java.

getYmax()

```
float visualization.ChartData.getYmax ( )
```

Returns

the greatest y coordinate of the points.

Definition at line 217 of file ChartData.java.

getYmin()

```
float visualization.ChartData.getYmin ( )
```

Returns

the smallest y coordinate of the points.

Definition at line 224 of file ChartData.java.

paint()

```
abstract void visualization.ChartData.paint (
    Graphics2D g ) [abstract]
```

Paints the element.

Parameters

<i>g</i>	the Graphics context in which to paint
----------	--

paintLegendElement()

```
abstract void visualization.ChartData.paintLegendElement (
    Graphics2D g,
    int x,
    int y ) [abstract]
```

Paints a legend representation for this element.

Parameters

<i>g</i>	the Graphics context in which to paint
<i>x</i>	the x coordinate of the drawing
<i>y</i>	the y coordinate of the drawing

setBoundariesToInclude()

```
void visualization.ChartData.setBoundariesToInclude (
    float x,
    float y )
```

Enlarges the boundaries if necessary, so that they include the coordinates.

Parameters

<i>x</i>	the x coordinate to include
<i>y</i>	the y coordinate to include

Definition at line 260 of file ChartData.java.

setColor()

```
void visualization.ChartData.setColor (
    Color color )
```

Sets the color in which to paint the element.

Parameters

<i>color</i>	the color in which to paint the element
--------------	---

Definition at line 278 of file ChartData.java.

setContainer()

```
void visualization.ChartData.setContainer (
    ChartContent container )
```

Sets the chart content to which this element belongs.

Parameters

<i>container</i>	the chart content to which this element belongs
------------------	---

Definition at line 289 of file ChartData.java.

setName()

```
void visualization.ChartData.setName (
    String name )
```

Sets the name of this element.

Parameters

<i>name</i>	the name of this element
-------------	--------------------------

Definition at line 300 of file ChartData.java.

6.17.4 Member Data Documentation**color**

```
Color visualization.ChartData.color [protected]
```

The color in which to paint the element.

Definition at line 107 of file ChartData.java.

container

```
ChartContent visualization.ChartData.container [protected]
```

The chart content to which the element belongs.

Can be null

Definition at line 97 of file ChartData.java.

data

```
List<Point2D.Float> visualization.ChartData.data [protected]
```

The list of points included in this element.

Definition at line 112 of file ChartData.java.

name

```
String visualization.ChartData.name [protected]
```

The name of the element.

Definition at line 102 of file ChartData.java.

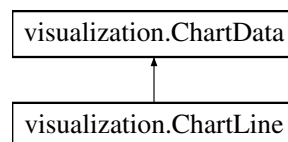
The documentation for this class was generated from the following file:

- src/visualization/[ChartData.java](#)

6.18 visualization.ChartLine Class Reference

A line to be displayed on a chart.

Inheritance diagram for visualization.ChartLine:

**Public Member Functions**

- [ChartLine](#) ()
- [ChartLine](#) (String [name](#))
- [ChartLine](#) (String [name](#), Color [color](#))
- void [paint](#) (Graphics2D g)
- void [paintLegendElement](#) (Graphics2D g, int x, int y)

Additional Inherited Members**6.18.1 Detailed Description**

A line to be displayed on a chart.

Definition at line 70 of file ChartLine.java.

6.18.2 Constructor & Destructor Documentation**ChartLine() [1/3]**

```
visualization.ChartLine.ChartLine ( )
```

Constructs an empty line.

Definition at line 76 of file ChartLine.java.

ChartLine() [2/3]

```
visualization.ChartLine.ChartLine (
    String name )
```

Constructs an empty line with a name.

Parameters

<i>name</i>	the name of the line
-------------	----------------------

Definition at line 87 of file ChartLine.java.

ChartLine() [3/3]

```
visualization.ChartLine.ChartLine (
    String name,
    Color color )
```

Constructs an empty line with a name and a color.

Parameters

<i>name</i>	the of the line
<i>color</i>	the color in which to paint the line

Definition at line 101 of file ChartLine.java.

6.18.3 Member Function Documentation**paint()**

```
void visualization.ChartLine.paint (
    Graphics2D g )
```

Paints the line.

Parameters

<i>g</i>	the Graphics context in which to paint
----------	--

Definition at line 114 of file ChartLine.java.

paintLegendElement()

```
void visualization.ChartLine.paintLegendElement (
    Graphics2D g,
    int x,
    int y )
```

Paints a legend representation for this line.

Parameters

<i>g</i>	the Graphics context in which to paint
<i>x</i>	the x coordinate of the drawing

Parameters

y	the y coordinate of the drawing
---	---------------------------------

Definition at line 143 of file ChartLine.java.

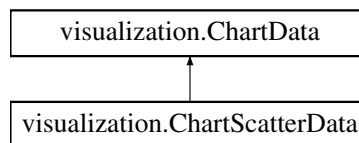
The documentation for this class was generated from the following file:

- src/visualization/[ChartLine.java](#)

6.19 visualization.ChartScatterData Class Reference

A collection of points to be displayed in a scatter chart.

Inheritance diagram for visualization.ChartScatterData:

**Public Member Functions**

- [ChartScatterData](#) ()
- [ChartScatterData](#) (String [name](#))
- [ChartScatterData](#) (String [name](#), Color [color](#))
- void [paint](#) (Graphics2D g)
- void [paintLegendElement](#) (Graphics2D g, int x, int y)

Additional Inherited Members

6.19.1 Detailed Description

A collection of points to be displayed in a scatter chart.

Definition at line 71 of file ChartScatterData.java.

6.19.2 Constructor & Destructor Documentation

ChartScatterData() [1/3]

```
visualization.ChartScatterData.ChartScatterData ( )
```

Constructs an empty scatter collection.

Definition at line 77 of file ChartScatterData.java.

ChartScatterData() [2/3]

```
visualization.ChartScatterData.ChartScatterData (
    String name )
```

Constructs an empty scatter collection with a name.

Parameters

<i>name</i>	the name of the collection
-------------	----------------------------

Definition at line 88 of file ChartScatterData.java.

ChartScatterData() [3/3]

```
visualization.ChartScatterData.ChartScatterData (
    String name,
    Color color )
```

Constructs an empty scatter collection with a name and a color.

Parameters

<i>name</i>	the name of the collection
<i>color</i>	the color in which to paint the collection

Definition at line 102 of file ChartScatterData.java.

6.19.3 Member Function Documentation**paint()**

```
void visualization.ChartScatterData.paint (
    Graphics2D g )
```

Paints the collection.

Parameters

<i>g</i>	the Graphics context in which to paint
----------	--

Definition at line 116 of file ChartScatterData.java.

paintLegendElement()

```
void visualization.ChartScatterData.paintLegendElement (
    Graphics2D g,
    int x,
    int y )
```

Paints a legend representation for this collection.

Parameters

<i>g</i>	the Graphics context in which to paint
<i>x</i>	the x coordinate of the drawing
<i>y</i>	the y coordinate of the drawing

Definition at line 143 of file ChartScatterData.java.

The documentation for this class was generated from the following file:

- [src/visualization/ChartScatterData.java](#)

6.20 visualization.DataFileReader Class Reference

Provides static method to read a simple data file and build a chart element from it.

Static Public Member Functions

- static void [buildDataSet](#) (File file, [ChartData](#) data) throws IOException

6.20.1 Detailed Description

Provides static method to read a simple data file and build a chart element from it.

Definition at line 71 of file DataFileReader.java.

6.20.2 Member Function Documentation

buildDataSet()

```
static void visualization.DataFileReader.buildDataSet (
    File file,
    ChartData data ) throws IOException [static]
```

Adds points to a chart element with the coordinate included in a file. The file must contain two numbers per line, the first being the x coordinate and the second the y coordinate. Commented lines beginning with # are ignored, as well as badly formatted lines.

Parameters

<i>file</i>	the file containing the data
<i>data</i>	the chart element to which the points must be added

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 87 of file DataFileReader.java.

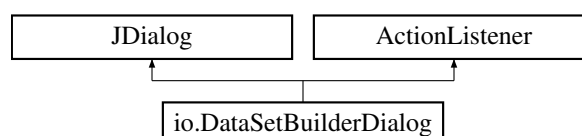
The documentation for this class was generated from the following file:

- [src/visualization/DataFileReader.java](#)

6.21 io.DataSetBuilderDialog Class Reference

A dialog box that allows the user to add custom data to a chart from a file.

Inheritance diagram for io.DataSetBuilderDialog:



Public Member Functions

- void `actionPerformed` (ActionEvent evt)

Static Public Member Functions

- static `ChartData showDialog` (Frame parent)

Static Protected Attributes

- static final int `MIN_LABEL_WIDTH`
A constant indicating the minimum width of field labels.
- static final int `MIN_FIELD_WIDTH`
A constant indicating the minimum width of text fields.

6.21.1 Detailed Description

A dialog box that allows the user to add custom data to a chart from a file.

The dialog contains fields to indicate the path of the file, the display name of the data, its type of display (line or points, color).

Definition at line 98 of file DataSetBuilderDialog.java.

6.21.2 Member Function Documentation

`actionPerformed()`

```
void io.DataSetBuilderDialog.actionPerformed (
   (ActionEvent evt )
```

Action performed when the user clicks on a button

Parameters

<code>evt</code>	the triggering event
------------------	----------------------

Definition at line 198 of file DataSetBuilderDialog.java.

`showDialog()`

```
static ChartData io.DataSetBuilderDialog.showDialog (
    Frame parent ) [static]
```

Shows a dialog box that allows the user to add custom to a chart from a file

Parameters

<code>parent</code>	the parent frame of the dialog box
---------------------	------------------------------------

Returns

a chart element

Definition at line 109 of file DataSetBuilderDialog.java.

6.21.3 Member Data Documentation

MIN_FIELD_WIDTH

```
final int io.DataSetBuilderDialog.MIN_FIELD_WIDTH [static], [protected]
```

A constant indicating the minimum width of text fields.

Definition at line 134 of file DataSetBuilderDialog.java.

MIN_LABEL_WIDTH

```
final int io.DataSetBuilderDialog.MIN_LABEL_WIDTH [static], [protected]
```

A constant indicating the minimum width of field labels.

Definition at line 129 of file DataSetBuilderDialog.java.

The documentation for this class was generated from the following file:

- [src/io/DataSetBuilderDialog.java](#)

6.22 model.definition.Definition_1D Class Reference

This class provides a static method for generating the configuration used by FullSWOF_1D parameters files.

Static Public Member Functions

- static [Node configuration](#) ()
- static void [checkFile](#) ()
- static void [checkIsValidTable](#) ()

6.22.1 Detailed Description

This class provides a static method for generating the configuration used by FullSWOF_1D parameters files.

Definition at line 77 of file Definition_1D.java.

6.22.2 Member Function Documentation

checkFile()

```
static void model.definition.Definition_1D.checkFile ( ) [static]
```

Definition at line 129 of file Definition_1D.java.

checkIsValidTable()

```
static void model.definition.Definition_1D.checkIsValidTable ( ) [static]
```

Definition at line 132 of file Definition_1D.java.

configuration()

```
static Node model.definition.Definition_1D.configuration ( ) [static]
```

Definition at line 112 of file Definition_1D.java.

The documentation for this class was generated from the following file:

- src/model/definition/[Definition_1D.java](#)

6.23 model.definition.Definition_2D Class Reference

This class provides a static method for generating the configuration used by FullSWOF_2D parameters files.

Static Public Member Functions

- static [Node](#) [configuration](#) ()
- static void [checkFile](#) ()
Check the formules.
- static void [checkIsValidTable](#) ()
check table
- static void [updateChangeEvent](#) ([Node](#) node)
Update the dropdown list.

6.23.1 Detailed Description

This class provides a static method for generating the configuration used by FullSWOF_2D parameters files.

Definition at line 77 of file Definition_2D.java.

6.23.2 Member Function Documentation**checkFile()**

```
static void model.definition.Definition_2D.checkFile ( ) [static]
```

Check the formules.

Definition at line 140 of file Definition_2D.java.

checkIsValidTable()

```
static void model.definition.Definition_2D.checkIsValidTable ( ) [static]
```

check table

Definition at line 147 of file Definition_2D.java.

configuration()

```
static Node model.definition.Definition_2D.configuration ( ) [static]
```

Returns

the configuration used by FullSWOF_2D parameters files

Definition at line 118 of file Definition_2D.java.

updateChangeEvent()

```
static void model.definition.Definition_2D.updateChangeEvent (
    Node node ) [static]
    Update the dropdown list.
```

Parameters

<i>node</i>	
-------------	--

Definition at line 155 of file Definition_2D.java.

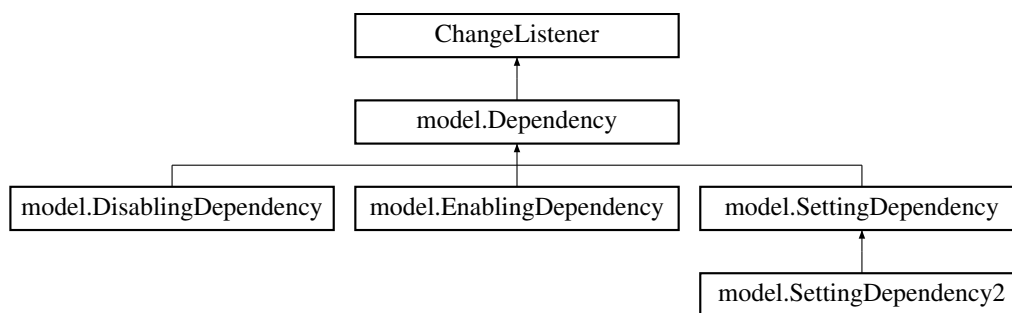
The documentation for this class was generated from the following file:

- src/model/definition/[Definition_2D.java](#)

6.24 model.Dependency Class Reference

A dependency is a binary relationship between external nodes.

Inheritance diagram for model.Dependency:

**Public Member Functions**

- [Dependency](#) ([ExternalNode](#) master, [ExternalNode](#) slave, String [targetValue](#))
- [ExternalNode](#) [getMaster](#) ()
- [ExternalNode](#) [getSlave](#) ()
- Object [getTargetValue](#) ()
- void [setEnabled](#) (boolean enabled)
- abstract boolean [isRespected](#) ()
- abstract void [resolve](#) ()
- void [stateChanged](#) ([ChangeEvent](#) evt)

Package Attributes

- [ExternalNode](#) [master](#)
The master node.
- [ExternalNode](#) [slave](#)
The slave node.
- String [targetValue](#)
The target value of the dependency.

6.24.1 Detailed Description

A dependency is a binary relationship between external nodes.

Every time the value of the master node is changed, it is compared to the target value of the dependency. If the new value of the master node is equal to that target value, a reaction is triggered on the slave node, which depends on the particular implementation of the dependency. Dependencies are very useful to use with a multiple choice parameter as the master node, where a certain choice will trigger changes on other parameters, such as disabling or enabling them.

Definition at line 76 of file `Dependency.java`.

6.24.2 Constructor & Destructor Documentation

Dependency()

```
model.Dependency.Dependency (
    ExternalNode master,
    ExternalNode slave,
    String targetValue )
```

Constructs a dependency between the master and slave node for a target value. The newly constructed dependency is automatically activated.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency

Definition at line 107 of file `Dependency.java`.

6.24.3 Member Function Documentation

getMaster()

```
ExternalNode model.Dependency.getMaster ( )
```

Returns

the master node of the dependency

Definition at line 119 of file `Dependency.java`.

getSlave()

```
ExternalNode model.Dependency.getSlave ( )
```

Returns

the slave node of the dependency

Definition at line 126 of file `Dependency.java`.

getTargetValue()

```
Object model.Dependency.getTargetValue ( )
```

Returns

the target value of the dependency

Definition at line 133 of file Dependency.java.

isRespected()

```
abstract boolean model.Dependency.isRespected ( ) [abstract]
```

Returns true if the dependency is respected, which should be the case after the [resolve\(\)](#) method has been invoked. If the master node value is different from the target value, this method should always return true. Other case depend on the particular implementation of the dependency

Returns

true if the dependency is respected

resolve()

```
abstract void model.Dependency.resolve ( ) [abstract]
```

The reaction triggered when the master node is set to the target value. This action is totally different for each type of dependency

setEnabled()

```
void model.Dependency.setEnabled (
    boolean enabled )
```

Sets whether or not this dependency is enabled

Parameters

<i>enabled</i>	true if this dependency should be enabled, false otherwise
----------------	--

Definition at line 142 of file Dependency.java.

stateChanged()

```
void model.Dependency.stateChanged (
    ChangeEvent evt )
```

Called when the master node fires a change event

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 177 of file Dependency.java.

6.24.4 Member Data Documentation

master

`ExternalNode` `model.Dependency.master` [package]

The master node.

Every time the value of this node is set equal to the target value of the dependency, the dependency's resolution will be triggered.

Definition at line 83 of file `Dependency.java`.

slave

`ExternalNode` `model.Dependency.slave` [package]

The slave node.

This node is the target of the dependency's resolution.

Definition at line 88 of file `Dependency.java`.

targetValue

`String` `model.Dependency.targetValue` [package]

The target value of the dependency.

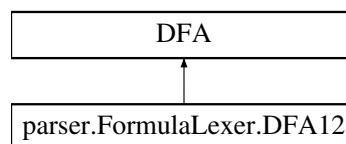
Definition at line 93 of file `Dependency.java`.

The documentation for this class was generated from the following file:

- `src/model/Dependency.java`

6.25 parser.FormulaLexer.DFA12 Class Reference

Inheritance diagram for `parser.FormulaLexer.DFA12`:



Public Member Functions

- `DFA12` (BaseRecognizer recognizer)
- `String` `getDescription` ()

6.25.1 Detailed Description

Definition at line 1567 of file `FormulaLexer.java`.

6.25.2 Constructor & Destructor Documentation

DFA12()

```
parser.FormulaLexer.DFA12.DFA12 (
    BaseRecognizer recognizer )
```

Definition at line 1569 of file FormulaLexer.java.

6.25.3 Member Function Documentation**getDescription()**

```
String parser.FormulaLexer.DFA12.getDescription ( )
```

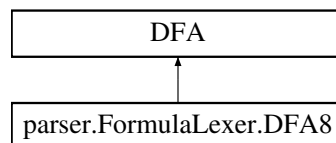
Definition at line 1581 of file FormulaLexer.java.

The documentation for this class was generated from the following file:

- src/parser/[FormulaLexer.java](#)

6.26 parser.FormulaLexer.DFA8 Class Reference

Inheritance diagram for parser.FormulaLexer.DFA8:

**Public Member Functions**

- [DFA8](#) (BaseRecognizer recognizer)
- String [getDescription](#) ()

6.26.1 Detailed Description

Definition at line 1497 of file FormulaLexer.java.

6.26.2 Constructor & Destructor Documentation**DFA8()**

```
parser.FormulaLexer.DFA8.DFA8 (
    BaseRecognizer recognizer )
```

Definition at line 1499 of file FormulaLexer.java.

6.26.3 Member Function Documentation

getDescription()

`String parser.FormulaLexer.DFA8.getDescription ()`

Definition at line 1511 of file `FormulaLexer.java`.

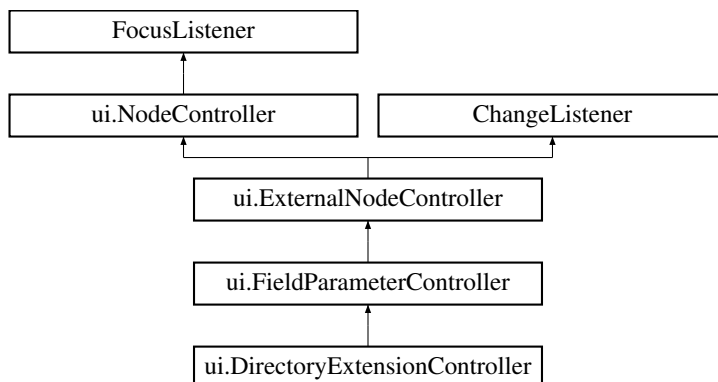
The documentation for this class was generated from the following file:

- `src/parser/FormulaLexer.java`

6.27 ui.DirectoryExtensionController Class Reference

A controller for a directory extension parameter.

Inheritance diagram for `ui.DirectoryExtensionController`:

**Public Member Functions**

- `DirectoryExtensionController` (`DirectoryExtensionParameter` node)
- boolean `validate` (File mainDirectory)

Additional Inherited Members**6.27.1 Detailed Description**

A controller for a directory extension parameter.

The view provided is similar to that of a `FieldParameterController` but the validation method includes the creation of a folder.

Definition at line 77 of file `DirectoryExtensionController.java`.

6.27.2 Constructor & Destructor Documentation**DirectoryExtensionController()**

`ui.DirectoryExtensionController.DirectoryExtensionController (`
 `DirectoryExtensionParameter node)`

Constructs a controller for a directory extension parameter node.

Parameters

<i>node</i>	the node to be controlled
-------------	---------------------------

Definition at line 86 of file `DirectoryExtensionController.java`.

6.27.3 Member Function Documentation

validate()

```
boolean ui.DirectoryExtensionController.validate (
    File mainDirectory )
```

Creates a directory named "Outputs"+this.model.getValue() under mainDirectory, unless this directory already existed.

Parameters

<i>mainDirectory</i>	the working project directory
----------------------	-------------------------------

Returns

true if the directory already existed or was successfully created.

Exceptions

<i>IllegalStateException</i>	if mainDirectory is not a directory
------------------------------	-------------------------------------

Definition at line 99 of file DirectoryExtensionController.java.

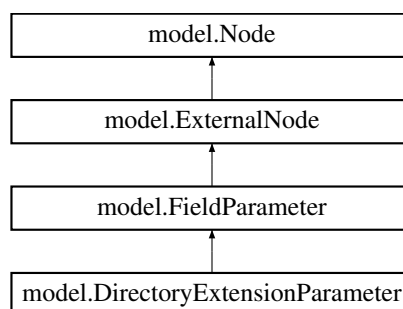
The documentation for this class was generated from the following file:

- [src/ui/DirectoryExtensionController.java](#)

6.28 model.DirectoryExtensionParameter Class Reference

A directory extension parameter is a special implementation of [ExternalNode](#), which is typically used only once in a configuration tree.

Inheritance diagram for model.DirectoryExtensionParameter:



Public Member Functions

- [DirectoryExtensionParameter](#) (String [name](#), String [tag](#))
- [DirectoryExtensionParameter](#) (String [name](#), String [tag](#), String [description](#))
- boolean [isValid](#) ()
- [NodeController](#) [setUpController](#) ()

Additional Inherited Members

6.28.1 Detailed Description

A directory extension parameter is a special implementation of [ExternalNode](#), which is typically used only once in a configuration tree.

It is used to indicate the suffix of the outputs folder used by FullSWOF, which will be named "`↵ Outputs`" + `this.value`.

Definition at line 72 of file `DirectoryExtensionParameter.java`.

6.28.2 Constructor & Destructor Documentation

DirectoryExtensionParameter() [1/2]

```
model.DirectoryExtensionParameter.DirectoryExtensionParameter (
    String name,
    String tag )
```

Constructs a directory extension parameter with the provided name and tag, and no description.

Parameters

<i>name</i>	the name of the parameter
<i>tag</i>	the tag of the parameter

Definition at line 84 of file `DirectoryExtensionParameter.java`.

DirectoryExtensionParameter() [2/2]

```
model.DirectoryExtensionParameter.DirectoryExtensionParameter (
    String name,
    String tag,
    String description )
```

Constructs a directory extension parameter with the provided name, tag and description.

Parameters

<i>name</i>	the name of the parameter
<i>tag</i>	the tag of the parameter
<i>description</i>	a description of the parameter

Definition at line 100 of file `DirectoryExtensionParameter.java`.

6.28.3 Member Function Documentation

isValid()

```
boolean model.DirectoryExtensionParameter.isValid ( )
```

Returns false if any of the node's dependencies are not respected or if the value contains `/`, `\`, `.` or `:` which are forbidden to use in a directory name.

Note that other characters might be forbidden on some platforms, especially windows, and that the white-space character is usually discouraged, but this method will not check the presence of these characters.

Definition at line 117 of file DirectoryExtensionParameter.java.

setUpController()

```
NodeController model.DirectoryExtensionParameter.setUpController ( )
```

Returns

the node controller associated with this type of parameter.

Definition at line 138 of file DirectoryExtensionParameter.java.

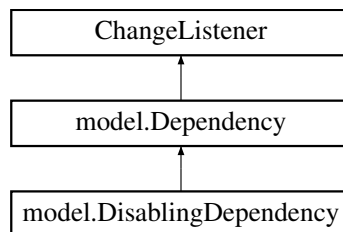
The documentation for this class was generated from the following file:

- src/model/[DirectoryExtensionParameter.java](#)

6.29 model.DisablingDependency Class Reference

A disabling dependency is used to disable the slave node when the master node is set to the target value.

Inheritance diagram for model.DisablingDependency:



Public Member Functions

- [DisablingDependency](#) ([ExternalNode](#) master, [ExternalNode](#) slave, String targetValue)
- boolean [isRespected](#) ()
- void [resolve](#) ()
- String [toString](#) ()

Additional Inherited Members

6.29.1 Detailed Description

A disabling dependency is used to disable the slave node when the master node is set to the target value.

Note that the slave node will not be automatically enabled if the master node value changes later. An enabling dependency should therefore be added on the master node for each other possible value. Without this, this user will have no mean to re-enable the parameters which was disabled.

Definition at line 73 of file DisablingDependency.java.

6.29.2 Constructor & Destructor Documentation

DisablingDependency()

```
model.DisablingDependency.DisablingDependency (
    ExternalNode master,
    ExternalNode slave,
    String targetValue )
```

Constructs a disabling dependency between the master and slave node for a target value.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency

Definition at line 87 of file DisablingDependency.java.

6.29.3 Member Function Documentation**isRespected()**

```
boolean model.DisablingDependency.isRespected ( )
```

Returns

false if the master node value is equal to the target value and the slave node is enabled.

Definition at line 97 of file DisablingDependency.java.

resolve()

```
void model.DisablingDependency.resolve ( )
```

Disables the slave node if the master node is equal to the target value.

Definition at line 110 of file DisablingDependency.java.

toString()

```
String model.DisablingDependency.toString ( )
```

Definition at line 116 of file DisablingDependency.java.

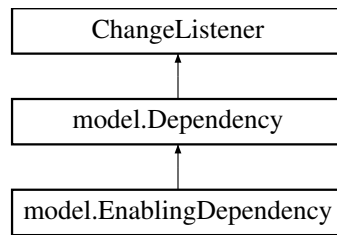
The documentation for this class was generated from the following file:

- src/model/[DisablingDependency.java](#)

6.30 model.EnablingDependency Class Reference

An enabling dependency is used to enable the slave node when the master node is set to the target value.

Inheritance diagram for model.EnablingDependency:



Public Member Functions

- [EnablingDependency](#) ([ExternalNode](#) master, [ExternalNode](#) slave, String [targetValue](#))
- boolean [isRespected](#) ()
- void [resolve](#) ()
- String [toString](#) ()

Additional Inherited Members

6.30.1 Detailed Description

An enabling dependency is used to enable the slave node when the master node is set to the target value.

All nodes are created enabled by default, so you need to create an enabling dependency only if a disabling dependency has been added to the same slave node.

Definition at line 72 of file `EnablingDependency.java`.

6.30.2 Constructor & Destructor Documentation

EnablingDependency()

```

model.EnablingDependency.EnablingDependency (
    ExternalNode master,
    ExternalNode slave,
    String targetValue )
  
```

Constructs an enabling dependency between the master and slave node for a target value.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency

Definition at line 86 of file `EnablingDependency.java`.

6.30.3 Member Function Documentation

isRespected()

```

boolean model.EnablingDependency.isRespected ( )
  
```

Returns

false if the master node value is equal to the target value and the slave node is disabled.

Definition at line 96 of file EnablingDependency.java.

resolve()

```
void model.EnablingDependency.resolve ( )
```

Enables the slave node if the master node is equal to the target value.

Definition at line 109 of file EnablingDependency.java.

toString()

```
String model.EnablingDependency.toString ( )
```

Definition at line 115 of file EnablingDependency.java.

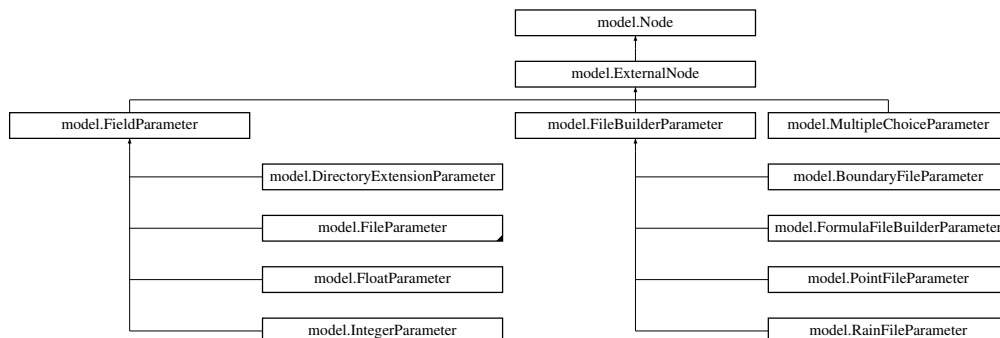
The documentation for this class was generated from the following file:

- src/model/[EnablingDependency.java](#)

6.31 model.ExternalNode Class Reference

An external node in the tree model, typically a FullSWOF parameter.

Inheritance diagram for model.ExternalNode:

**Public Member Functions**

- [ExternalNode](#) (String [name](#), String [tag](#))
- [ExternalNode](#) (String [name](#), String [tag](#), String [description](#))
- void [addChangeListener](#) (ChangeListener c)
- boolean [fromFile](#) (File file) throws IOException
- List< [Dependency](#) > [getDependencies](#) ()
- void [removeDependency](#) ([Dependency](#) d)
- String [getValue](#) ()
- boolean [isEnabled](#) ()
- abstract boolean [isValid](#) ()
- void [removeChangeListener](#) (ChangeListener c)
- void [setEnabled](#) (boolean [enabled](#))
- void [setValue](#) (String newValue)
- String [toFile](#) (boolean verbose)
- void [updateChangeEvent](#) (String [tag](#))

Protected Member Functions

- boolean `addDependency` (`Dependency` d)
- void `fireChangeEvent` ()

Package Attributes

- String `tag`
The tag used in the parameters.txt file to describe this parameter.
- String `value`
The value of the parameter.
- boolean `enabled`
Indicates if the node is enabled.
- List< `Dependency` > `dependencies`
The list of dependencies linked to that node.
- EventListenerList `listenerList`
The list of objects listening to event fired by this node.
- ChangeEvent `changeEvent`
Used to lazily create change events.

Additional Inherited Members

6.31.1 Detailed Description

An external node in the tree model, typically a FullSWOF parameter.

An external node does not have any child nodes.

Definition at line 80 of file `ExternalNode.java`.

6.31.2 Constructor & Destructor Documentation

ExternalNode() [1/2]

```
model.ExternalNode.ExternalNode (
    String name,
    String tag )
```

Constructs an external node with the provided name and tag, and no description.

Parameters

<i>name</i>	the name of the node.
<i>tag</i>	the tag of the node.

Definition at line 133 of file `ExternalNode.java`.

ExternalNode() [2/2]

```
model.ExternalNode.ExternalNode (
    String name,
    String tag,
    String description )
```

Constructs an external node with the provided name, tag and description.

Parameters

<i>name</i>	the name of the node.
<i>tag</i>	the tag of the node.
<i>description</i>	a description of the node.

Definition at line 149 of file ExternalNode.java.

6.31.3 Member Function Documentation**addChangeListener()**

```
void model.ExternalNode.addChangeListener (
    ChangeListener c )
```

Adds a change listener to the node. Change listener should be notified of any change on the node's value or enabled attribute.

Parameters

<i>c</i>	the change listener to be added.
----------	----------------------------------

Definition at line 162 of file ExternalNode.java.

addDependency()

```
boolean model.ExternalNode.addDependency (
    Dependency d ) [protected]
```

Adds a dependency to this node's dependencies list. This list is useful only to make sure that dependencies on that node are respected.

Parameters

<i>d</i>	the dependency to be added
----------	----------------------------

Returns

true if the dependency was successfully added.

Exceptions

<i>IllegalStateException</i>	if the dependency's master is not the node itself.
------------------------------	--

Definition at line 177 of file ExternalNode.java.

fireChangeEvent()

```
void model.ExternalNode.fireChangeEvent ( ) [protected]
```

Invoked when the value of the node is changed.

See also

`javax.swing.event.EventListenerList`

Definition at line 336 of file `ExternalNode.java`.

fromFile()

```
boolean model.ExternalNode.fromFile (
    File file ) throws IOException
```

Attempts to find the node's tag in the file and interpret the string after it as the node's new value. If the next non-whitespace character after the tag is a newline, the value is not changed. The format for a FullSWOF parameter tag is `<tag>:: value` but for implementation reasons the colon character is treated the same as a whitespace character by this method, so the number of colons does not actually matter, nor does the number of whitespace characters between the tag and its value.

Parameters

<i>file</i>	the file to be read
-------------	---------------------

Returns

true if the node's value was found and changed.

Exceptions

<i>IOException</i>	if a problem occurred while reading the file, such as the file not being found.
--------------------	---

Definition at line 201 of file `ExternalNode.java`.

getDependencies()

```
List<Dependency> model.ExternalNode.getDependencies ( )
```

Returns

the list of dependencies linked to that node.

Definition at line 228 of file `ExternalNode.java`.

getValue()

```
String model.ExternalNode.getValue ( )
```

Returns

the value of the node.

Definition at line 246 of file `ExternalNode.java`.

isEnabled()

```
boolean model.ExternalNode.isEnabled ( )
```

Returns

true if the node is enabled.

Definition at line 253 of file ExternalNode.java.

isValid()

```
abstract boolean model.ExternalNode.isValid ( ) [abstract]
```

Each extending class must provide its own implementation of this method, which is what differentiate parameter types from each others.

Implementation note: this method should at least return false if any of the dependencies is not respected.

removeChangeListener()

```
void model.ExternalNode.removeChangeListener (
    ChangeListener c )
```

Removes a change listener from the change listeners list of this node.

Parameters

<i>c</i>	the change listener to be removed
----------	-----------------------------------

Definition at line 275 of file ExternalNode.java.

removeDependency()

```
void model.ExternalNode.removeDependency (
    Dependency d )
```

Removes a dependency from that node.

Parameters

<i>d</i>	the dependency to be removed
----------	------------------------------

Definition at line 239 of file ExternalNode.java.

setEnabled()

```
void model.ExternalNode.setEnabled (
    boolean enabled )
```

Enables or disables the node.

Parameters

<i>enabled</i>	true to enable the node, false to disable it
----------------	--

Definition at line 286 of file ExternalNode.java.

setValue()

```
void model.ExternalNode.setValue (
    String newValue )
```

Sets the value of this node. A change event will be fired and dependencies resolved only if the new value is different from the old.

Parameters

<i>newValue</i>	the new value of the node
-----------------	---------------------------

Definition at line 299 of file ExternalNode.java.

toFile()

```
String model.ExternalNode.toFile (
    boolean verbose )
```

Returns a string to be written in a parameters.txt file. The string will be of the form <tag>:: value followed by an newline character. The tag will be preceded by a description if verbose is set to true.

Parameters

<i>verbose</i>	indicates whether the file should include a description of the node
----------------	---

Returns

a string to be written in a parameters.txt file.

Definition at line 321 of file ExternalNode.java.

updateChangeEvent()

```
void model.ExternalNode.updateChangeEvent (
    String tag )
```

Check, if the tag sought is equal to this instance if true so there is an update events

Parameters

<i>tag</i>	the tag sought
------------	----------------

Definition at line 359 of file ExternalNode.java.

6.31.4 Member Data Documentation**changeEvent**

```
ChangeEvent model.ExternalNode.changeEvent [package]
```

Used to lazily create change events.

See also

`javax.swing.event.EventListenerList`

Definition at line 121 of file `ExternalNode.java`.

dependencies

`List<Dependency> model.ExternalNode.dependencies [package]`

The list of dependencies linked to that node.

This list is used only to make sure that dependencies on that node are respected.

See also

[Dependency](#)

Definition at line 106 of file `ExternalNode.java`.

enabled

`boolean model.ExternalNode.enabled [package]`

Indicates if the node is enabled.

Some parameters are disabled under specific conditions. A disabled node should always be valid. Note that its tag must still be written in the `parameters.txt` file, but its value can be left empty

Definition at line 98 of file `ExternalNode.java`.

listenerList

`EventListenerList model.ExternalNode.listenerList [package]`

The list of objects listening to event fired by this node.

This list typically contains the controller of the node

See also

`javax.swing.event.EventListenerList`

Definition at line 114 of file `ExternalNode.java`.

tag

`String model.ExternalNode.tag [package]`

The tag used in the `parameters.txt` file to describe this parameter.

Definition at line 85 of file `ExternalNode.java`.

value

`String model.ExternalNode.value [package]`

The value of the parameter.

Definition at line 90 of file `ExternalNode.java`.

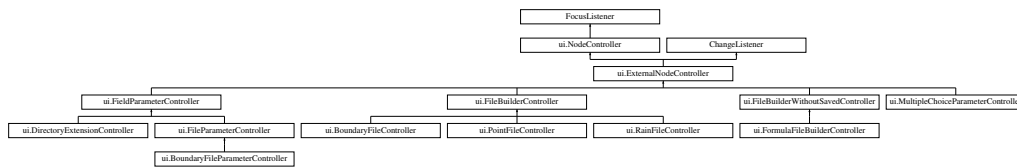
The documentation for this class was generated from the following file:

- `src/model/ExternalNode.java`

6.32 ui.ExternalNodeController Class Reference

A controller for an external node.

Inheritance diagram for ui.ExternalNodeController:



Public Member Functions

- [ExternalNodeController](#) ([ExternalNode model](#))
- void [focusLost](#) (FocusEvent e)
- void [stateChanged](#) (ChangeEvent e)
- boolean [validate](#) (File mainDirectory)

Package Functions

- abstract void [highlightView](#) ()
- abstract void [updateModel](#) ()
- abstract void [updateView](#) ()

Additional Inherited Members

6.32.1 Detailed Description

A controller for an external node.

Definition at line 72 of file ExternalNodeController.java.

6.32.2 Constructor & Destructor Documentation

ExternalNodeController()

```
ui.ExternalNodeController.ExternalNodeController (
    ExternalNode model )
```

Constructs a controller for an external node then instantiate a view for this node.

Parameters

<i>model</i>	the external node to be controlled
--------------	------------------------------------

Definition at line 83 of file ExternalNodeController.java.

6.32.3 Member Function Documentation

focusLost()

```
void ui.ExternalNodeController.focusLost (
    FocusEvent e )
```

Called every time the view loses the focus. The model is then updated.

Definition at line 95 of file ExternalNodeController.java.

highlightView()

```
abstract void ui.ExternalNodeController.highlightView ( ) [abstract], [package]
```

Brings the user attention on the view of this particular controller. This can imply grabbing the focus or coloring the view.

stateChanged()

```
void ui.ExternalNodeController.stateChanged (
    ChangeEvent e )
```

Called every time the model is changed. The view is then updated.

Definition at line 103 of file ExternalNodeController.java.

updateModel()

```
abstract void ui.ExternalNodeController.updateModel ( ) [abstract], [package]
```

Updates the model with information from the view.

updateView()

```
abstract void ui.ExternalNodeController.updateView ( ) [abstract], [package]
```

Updates the view with information from the model.

validate()

```
boolean ui.ExternalNodeController.validate (
    File mainDirectory )
```

Applies validation procedures to the node. This method is called when a project using this node is saved or run. The model is checked, and if it is not valid, its view is highlighted.

Definition at line 114 of file ExternalNodeController.java.

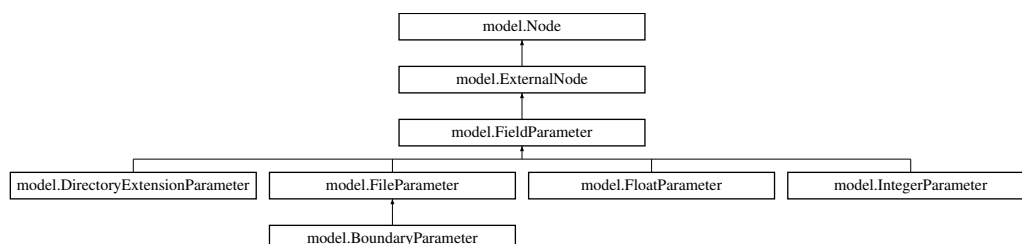
The documentation for this class was generated from the following file:

- [src/ui/ExternalNodeController.java](#)

6.33 model.FieldParameter Class Reference

This class provides the most permissive implementation of an external node, as any value will be considered a valid entry.

Inheritance diagram for model.FieldParameter:



Public Member Functions

- boolean `isValid` ()
- `NodeController` `setUpController` ()

Package Functions

- `FieldParameter` (String `name`, String `tag`)
- `FieldParameter` (String `name`, String `tag`, String `description`)

Additional Inherited Members

6.33.1 Detailed Description

This class provides the most permissive implementation of an external node, as any value will be considered a valid entry.

Definition at line 68 of file `FieldParameter.java`.

6.33.2 Constructor & Destructor Documentation

`FieldParameter()` [1/2]

```
model.FieldParameter.FieldParameter (
    String name,
    String tag ) [package]
```

Constructs a field parameter with the provided name and tag and, no description.

Parameters

<i>name</i>	
<i>tag</i>	

Definition at line 78 of file `FieldParameter.java`.

`FieldParameter()` [2/2]

```
model.FieldParameter.FieldParameter (
    String name,
    String tag,
    String description ) [package]
```

Constructs a field parameter with the provided name, tag and description.

Parameters

<i>name</i>	
<i>tag</i>	
<i>description</i>	

Definition at line 90 of file `FieldParameter.java`.

6.33.3 Member Function Documentation

isValid()

```
boolean model.FieldParameter.isValid ( )
```

Returns

true if all the node's dependencies are respected.

Definition at line 98 of file FieldParameter.java.

setUpController()

```
NodeController model.FieldParameter.setUpController ( )
```

Build a controller for this node.

Returns

the controller for this node.

See also

[ui.FieldParameterController](#)

Definition at line 115 of file FieldParameter.java.

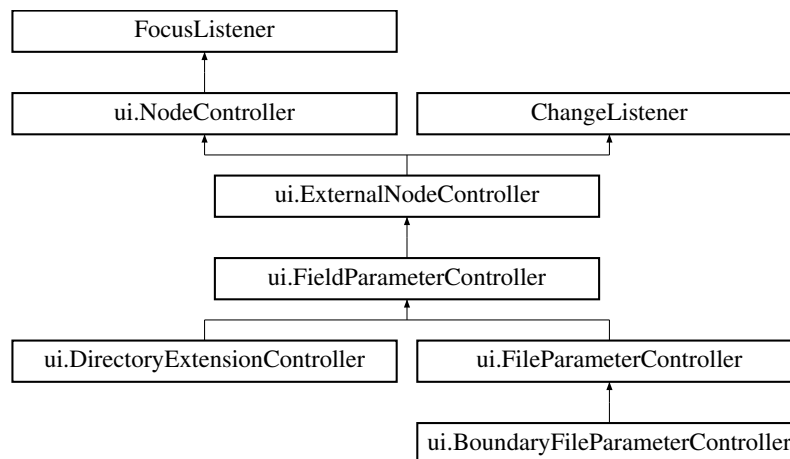
The documentation for this class was generated from the following file:

- [src/model/FieldParameter.java](#)

6.34 ui.FieldParameterController Class Reference

A controller for a field parameter.

Inheritance diagram for ui.FieldParameterController:

**Public Member Functions**

- [FieldParameterController](#) ([FieldParameter](#) node)

Package Functions

- void [highlightView](#) ()
- void [setUpView](#) ()
- void [updateModel](#) ()
- void [updateView](#) ()

Package Attributes

- JLabel [viewLabel](#)
The label of the view.
- JTextField [viewField](#)
The text field of the view.

Additional Inherited Members**6.34.1 Detailed Description**

A controller for a field parameter.

details This controller provides a view made of a label followed by a text field for input.

Definition at line 77 of file FieldParameterController.java.

6.34.2 Constructor & Destructor Documentation**FieldParameterController()**

```
ui.FieldParameterController.FieldParameterController (
    FieldParameter node )
```

Constructs a controller for a field parameter.

Parameters

<i>node</i>	the field parameter to be controlled
-------------	--------------------------------------

Definition at line 96 of file FieldParameterController.java.

6.34.3 Member Function Documentation**highlightView()**

```
void ui.FieldParameterController.highlightView ( ) [package]
```

Puts the focus on the text field and colors it in red.

Definition at line 105 of file FieldParameterController.java.

setUpView()

```
void ui.FieldParameterController.setUpView ( ) [package]
```

Sets up a view for this controller. The view is made of a label, followed by a text field on a single flowing line.

Definition at line 116 of file FieldParameterController.java.

updateModel()

```
void ui.FieldParameterController.updateModel ( ) [package]
```

Definition at line 132 of file FieldParameterController.java.

updateView()

```
void ui.FieldParameterController.updateView ( ) [package]
```

Definition at line 139 of file FieldParameterController.java.

6.34.4 Member Data Documentation**viewField**

```
JTextField ui.FieldParameterController.viewField [package]
```

The text field of the view.

Definition at line 87 of file FieldParameterController.java.

viewLabel

```
JLabel ui.FieldParameterController.viewLabel [package]
```

The label of the view.

Definition at line 82 of file FieldParameterController.java.

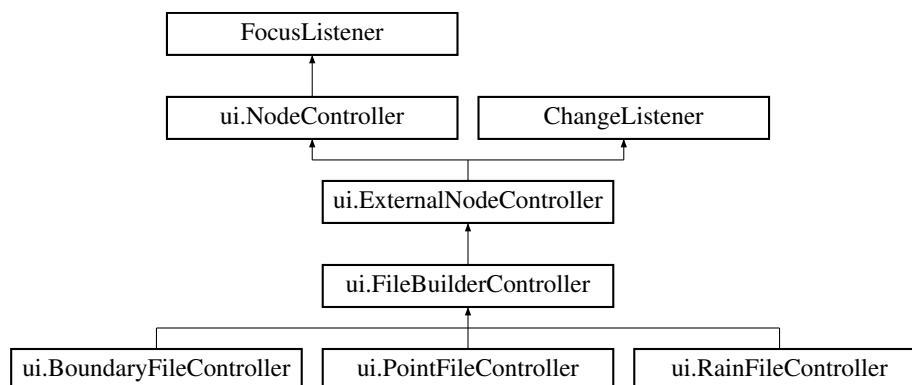
The documentation for this class was generated from the following file:

- [src/ui/FieldParameterController.java](#)

6.35 ui.FileBuilderController Class Reference

A controller for a file builder parameter.

Inheritance diagram for ui.FileBuilderController:

**Public Member Functions**

- [FileBuilderController](#) ([FileBuilderParameter](#) model)
- boolean [validate](#) (File mainDirectory)

Additional Inherited Members**6.35.1 Detailed Description**

A controller for a file builder parameter.

It records even if there are errors

Definition at line 72 of file FileBuilderController.java.

6.35.2 Constructor & Destructor Documentation

FileBuilderController()

```
ui.FileBuilderController.FileBuilderController (
    FileBuilderParameter model )
```

Constructs a controller for a file builder node.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 81 of file FileBuilderController.java.

6.35.3 Member Function Documentation

validate()

```
boolean ui.FileBuilderController.validate (
    File mainDirectory )
```

Builds a file in the 'Inputs' directory located in mainDirectory according to the specifications of the file builder node.

Parameters

<i>mainDirectory</i>	the project directory located above the 'Inputs' directory
----------------------	--

Returns

true if the file has been correctly written.

Definition at line 96 of file FileBuilderController.java.

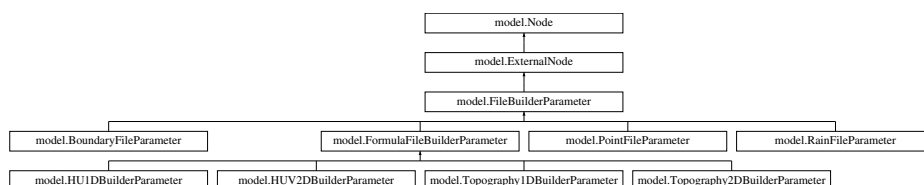
The documentation for this class was generated from the following file:

- [src/ui/FileBuilderController.java](#)

6.36 model.FileBuilderParameter Class Reference

A parameter used to create an annex file.

Inheritance diagram for model.FileBuilderParameter:



Public Member Functions

- [FileBuilderParameter](#) (String [name](#), String [fileName](#))
- boolean [fromFile](#) (File file) throws IOException

- String [toFile](#) (boolean verbose)
- String [getFileName](#) ()
- abstract String [getFileContent](#) ()
- String [getValue](#) ()
- void [setValue](#) (String newValue)
- boolean [addDependency](#) ([Dependency](#) d)
- boolean [isValidTable](#) ()

Package Attributes

- String [fileName](#)
The name of the file to be written.

Additional Inherited Members

6.36.1 Detailed Description

A parameter used to create an annex file.

Unlike other external nodes, a file builder parameter does not appear in the parameters.txt file. They are used to write an annex file such as a rain, topography or huv file, upon validation by the controller. Note that this external node does not have a value, so operations dealing with the node value (including dependencies) will throw an exception.

See also

[ui.FileBuilderController](#)

Definition at line 78 of file FileBuilderParameter.java.

6.36.2 Constructor & Destructor Documentation

FileBuilderParameter()

```
model.FileBuilderParameter.FileBuilderParameter (
    String name,
    String fileName )
```

Constructs a file builder with the provided name and file name.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file to be written

Definition at line 94 of file FileBuilderParameter.java.

6.36.3 Member Function Documentation

addDependency()

```
boolean model.FileBuilderParameter.addDependency (
    Dependency d )
```

Unsupported operation

Exceptions

<i>UnsupportedOperationException</i>	
--------------------------------------	--

Definition at line 165 of file FileBuilderParameter.java.

fromFile()

```
boolean model.FileBuilderParameter.fromFile (
    File file ) throws IOException
```

Simply returns true, since the parameters.txt file is irrelevant for this parameter.

Parameters

<i>file</i>	not used
-------------	----------

Returns

true.

Definition at line 109 of file FileBuilderParameter.java.

getFileContent()

```
abstract String model.FileBuilderParameter.getFileContent ( ) [abstract]
```

Returns

the string to be written in the file.

getFileName()

```
String model.FileBuilderParameter.getFileName ( )
```

Returns

the name of the file to be written.

Definition at line 130 of file FileBuilderParameter.java.

getValue()

```
String model.FileBuilderParameter.getValue ( )
    Unsupported operation
```

Exceptions

<i>UnsupportedOperationException</i>	
--------------------------------------	--

Definition at line 145 of file FileBuilderParameter.java.

isValidTable()

```
boolean model.FileBuilderParameter.isValidTable ( )
```

Returns true if the value is valid. The value can be enabled or disabled

Returns

true if the value is valid.

Definition at line 176 of file FileBuilderParameter.java.

setValue()

```
void model.FileBuilderParameter.setValue (
    String newValue )
```

Unsupported operation

Exceptions

<i>UnsupportedOperationException</i>	
--------------------------------------	--

Definition at line 155 of file FileBuilderParameter.java.

toFile()

```
String model.FileBuilderParameter.toFile (
    boolean verbose )
```

Returns an empty string, since this parameter does not appear in the parameters.txt file.

Parameters

<i>verbose</i>	not used
----------------	----------

Returns

an empty string.

Definition at line 123 of file FileBuilderParameter.java.

6.36.4 Member Data Documentation**fileName**

```
String model.FileBuilderParameter.fileName [package]
```

The name of the file to be written.

Definition at line 83 of file FileBuilderParameter.java.

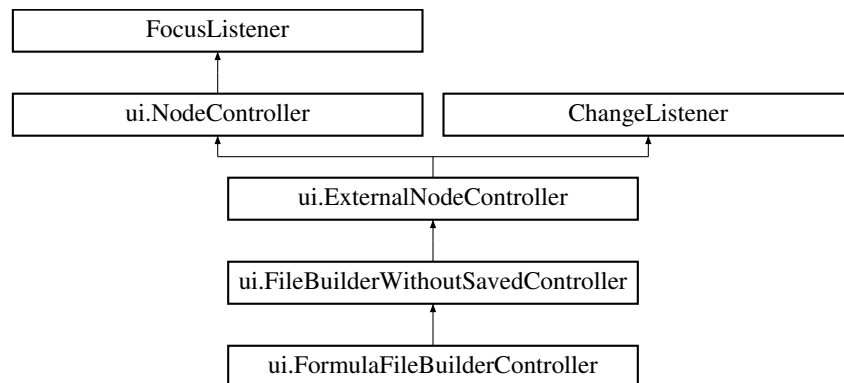
The documentation for this class was generated from the following file:

- src/model/[FileBuilderParameter.java](#)

6.37 ui.FileBuilderWithoutSavedController Class Reference

A controller for a file builder parameter.

Inheritance diagram for ui.FileBuilderWithoutSavedController:



Public Member Functions

- [FileBuilderWithoutSavedController](#) ([FileBuilderParameter](#) model)
- boolean [validate](#) (File mainDirectory)

Additional Inherited Members

6.37.1 Detailed Description

A controller for a file builder parameter.

It doesn't save if there are errors

Definition at line 71 of file FileBuilderWithoutSavedController.java.

6.37.2 Constructor & Destructor Documentation

FileBuilderWithoutSavedController()

```
ui.FileBuilderWithoutSavedController.FileBuilderWithoutSavedController (
    FileBuilderParameter model )
```

Constructs a controller for a file builder node.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 80 of file FileBuilderWithoutSavedController.java.

6.37.3 Member Function Documentation

validate()

```
boolean ui.FileBuilderWithoutSavedController.validate (
    File mainDirectory )
```

Builds a file in the 'Inputs' directory located in mainDirectory according to the specifications of the file builder node.

Parameters

<i>mainDirectory</i>	the project directory located above the 'Inputs' directory
----------------------	--

Returns

true if the file has been correctly written.

Definition at line 95 of file FileBuilderWithoutSavedController.java.

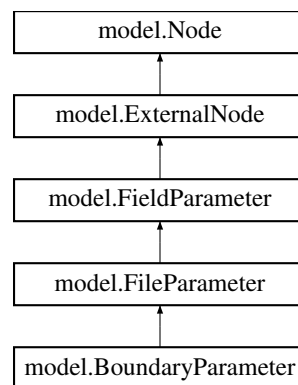
The documentation for this class was generated from the following file:

- [src/ui/FileBuilderWithoutSavedController.java](#)

6.38 model.FileParameter Class Reference

A parameter used to store a file pathname.

Inheritance diagram for model.FileParameter:



Public Member Functions

- [FileParameter](#) (String *name*, String *tag*)
- [FileParameter](#) (String *name*, String *tag*, [InputFileVisualizer](#) *visualizer*)
- [InputFileVisualizer](#) [getVisualizer](#) ()
- void [setVisualizer](#) ([InputFileVisualizer](#) *visualizer*)
- [FileParameter](#) (String *name*, String *tag*, String *description*)
- [FileParameter](#) (String *name*, String *tag*, String *description*, [InputFileVisualizer](#) *visualizer*)
- boolean [fromFile](#) (File *file*) throws IOException
- boolean [isValid](#) ()
- [NodeController](#) [setUpController](#) ()
- String [toFile](#) (boolean *verbose*)

Package Attributes

- [InputFileVisualizer](#) *visualizer*

The tool used to get a quick visualization of the input file.

Additional Inherited Members

6.38.1 Detailed Description

A parameter used to store a file pathname.

The file must be declared by its absolute pathname and must exist at that location. You can optionally specify a visualization tool for this parameter. In this case, the view will provide a way for the user to get a quick visualization of the file content (for example a chart).

Definition at line 81 of file FileParameter.java.

6.38.2 Constructor & Destructor Documentation

FileParameter() [1/4]

```
model.FileParameter.FileParameter (
    String name,
    String tag )
```

Constructs a file parameter with the provided name and tag, and no description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node

Definition at line 98 of file FileParameter.java.

FileParameter() [2/4]

```
model.FileParameter.FileParameter (
    String name,
    String tag,
    InputFileVisualizer visualizer )
```

Constructs a file parameter with the provided name, tag, visualizer and no description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>visualizer</i>	the tool used to get a quick visualization of the input file

Definition at line 114 of file FileParameter.java.

FileParameter() [3/4]

```
model.FileParameter.FileParameter (
    String name,
    String tag,
    String description )
```

Constructs a file parameter with the provided name, tag and description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node

Definition at line 153 of file FileParameter.java.

FileParameter() [4/4]

```
model.FileParameter.FileParameter (
    String name,
    String tag,
    String description,
    InputFileVisualizer visualizer )
```

Constructs a file parameter with the provided name, tag, description and visualizer.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node
<i>visualizer</i>	the tool used to get a quick visualization of the input file

Definition at line 171 of file FileParameter.java.

6.38.3 Member Function Documentation**fromFile()**

```
boolean model.FileParameter.fromFile (
    File file ) throws IOException
```

Attempts to set the values of the parameter from a file. Since files are only designated by their name in parameters.txt files, the location of that file is supposed to be the parent directory of the parameters.txt file.

Parameters

<i>file</i>	the parameters file containing the value
-------------	--

Returns

true if the parameter was successfully set.

Exceptions

<i>IOException</i>	if a problem occurred while reading the file, such as the file not being found
--------------------	--

Definition at line 190 of file FileParameter.java.

getVisualizer()

```
InputFileVisualizer model.FileParameter.getVisualizer ( )
```

Returns the file visualizer, the tool used to get a quick visualization of the input file.

Returns

the file visualizer.

Definition at line 126 of file FileParameter.java.

isValid()

```
boolean model.FileParameter.isValid ( )
```

Returns true if all the node dependencies are respected and the designated file exists and is a file (as opposed to a directory).

Returns

true if all the node dependencies are respected and the designated file exists.

Definition at line 223 of file FileParameter.java.

setUpController()

```
NodeController model.FileParameter.setUpController ( )
```

Builds a controller for the node.

Returns

a node controller.

See also

[ui.FileParameterController](#)

Definition at line 243 of file FileParameter.java.

setVisualizer()

```
void model.FileParameter.setVisualizer (
    InputFileVisualizer visualizer )
```

Sets the file visualizer, the tool used to get a quick visualization of the input file.

Parameters

<i>visualizer</i>	the file visualizer
-------------------	---------------------

Definition at line 138 of file FileParameter.java.

toFile()

```
String model.FileParameter.toFile (
    boolean verbose )
```

Returns a string to be written in a parameters.txt file. The string will be of the form <tag>:: value. In this case, the value is only the file name, not the absolute pathname stored in the node.

Returns

a string.

Definition at line 255 of file FileParameter.java.

6.38.4 Member Data Documentation

visualizer

`InputFileVisualizer` `model.FileParameter.visualizer` [package]

The tool used to get a quick visualization of the input file.

Definition at line 86 of file FileParameter.java.

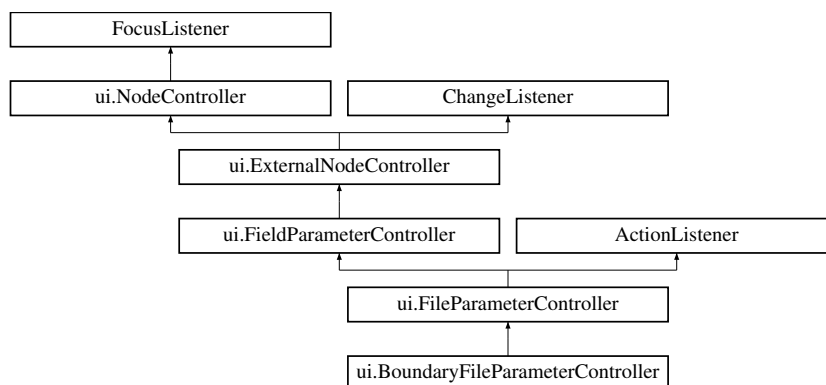
The documentation for this class was generated from the following file:

- `src/model/FileParameter.java`

6.39 ui.FileParameterController Class Reference

A controller for a file parameter node.

Inheritance diagram for ui.FileParameterController:



Public Member Functions

- `FileParameterController` (`FileParameter` model)
- void `actionPerformed` (`ActionEvent` evt)
- boolean `validate` (`File` mainDirectory)

Protected Member Functions

- void `copyFile` (`File` src, `File` dst) throws `IOException`
Copies the source file src if it exists to the destination dst.

Static Protected Attributes

- static `ResourceBundle` `messages`
The resource bundle containing the locale-specific strings displayed on the user interface.

Package Functions

- void `setUpView` ()
- void `updateModel` ()
- void `updateView` ()

Additional Inherited Members

6.39.1 Detailed Description

A controller for a file parameter node.

This controller can set up a view suited for file browsing. If the file parameter has a visualizer defined, the view enables the user to activate it.

Definition at line 94 of file `FileParameterController.java`.

6.39.2 Constructor & Destructor Documentation

`FileParameterController()`

```
ui.FileParameterController.FileParameterController (
    FileParameter model )
```

Constructs a controller for a file parameter node.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 121 of file `FileParameterController.java`.

6.39.3 Member Function Documentation

`actionPerformed()`

```
void ui.FileParameterController.actionPerformed (
    ActionEvent evt )
```

Called when the 'browse' button is clicked. Opens a file chooser.

Definition at line 129 of file `FileParameterController.java`.

`copyFile()`

```
void ui.FileParameterController.copyFile (
    File src,
    File dst ) throws IOException [protected]
```

Copies the source file `src` if it exists to the destination `dst`.

If the file does not exist, the method simply exits

Parameters

<i>src</i>	the file to be copied
<i>dst</i>	the destination of the copy

Exceptions

<i>IOException</i>	if an error occurs during the copy.
--------------------	-------------------------------------

Definition at line 282 of file FileParameterController.java.

setUpView()

```
void ui.FileParameterController.setUpView ( ) [package]
```

Sets up the view for this controller. The view is made of a label, followed by a text field and a 'browse' button that opens a file chooser. The three component are located on a single flowing line.

Definition at line 227 of file FileParameterController.java.

updateModel()

```
void ui.FileParameterController.updateModel ( ) [package]
```

Definition at line 256 of file FileParameterController.java.

updateView()

```
void ui.FileParameterController.updateView ( ) [package]
```

Definition at line 263 of file FileParameterController.java.

validate()

```
boolean ui.FileParameterController.validate (
    File mainDirectory )
```

Applies validation procedures to the node. This method is called when a project using this node is saved or run. The validation procedure for a file parameter includes copying the file designated by the model in mainDirectory if it is not already located there.

Parameters

<i>mainDirectory</i>	the directory in which to copy the file
----------------------	---

Returns

true if the file was successfully copied or was already in mainDirectory.

Exceptions

<i>IllegalStateException</i>	if mainDirectory is not a directory
------------------------------	-------------------------------------

Definition at line 191 of file FileParameterController.java.

6.39.4 Member Data Documentation

messages

```
ResourceBundle ui.FileParameterController.messages [static], [protected]
```

The resource bundle containing the locale-specific strings displayed on the user interface.

Definition at line 101 of file FileParameterController.java.

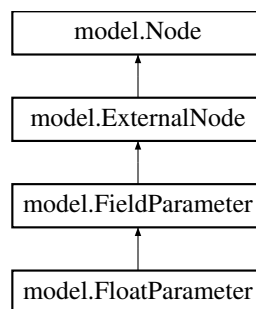
The documentation for this class was generated from the following file:

- [src/ui/FileParameterController.java](#)

6.40 model.FloatParameter Class Reference

A parameter with a floating point number value.

Inheritance diagram for model.FloatParameter:

**Public Member Functions**

- [FloatParameter](#) (String [name](#), String [tag](#))
- [FloatParameter](#) (String [name](#), String [tag](#), String [description](#))
- [FloatParameter](#) (String [name](#), String [tag](#), [Interval](#) valueInterval)
- [FloatParameter](#) (String [name](#), String [tag](#), String [description](#), [Interval](#) valueInterval)
- boolean [isValid](#) ()
- [NodeController](#) [setUpController](#) ()

Additional Inherited Members**6.40.1 Detailed Description**

A parameter with a floating point number value.

The value is still stored as a string, which is parsed to check whether it represents a numeric value. The acceptance interval of the value can be specified in the constructor or omitted, in which case any real number will be considered valid.

See also

`java.lang.Float.valueOf(String s)` to learn more about the lexical syntax rules for writing a floating point number as a string.

Definition at line 76 of file FloatParameter.java.

6.40.2 Constructor & Destructor Documentation

FloatParameter() [1/4]

```
model.FloatParameter.FloatParameter (
    String name,
    String tag )
```

Constructs a floating point number parameter with the provided name and tag, and no description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node

Definition at line 94 of file FloatParameter.java.

FloatParameter() [2/4]

```
model.FloatParameter.FloatParameter (
    String name,
    String tag,
    String description )
```

Constructs a floating point number parameter with the provided name, tag and description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node

Definition at line 110 of file FloatParameter.java.

FloatParameter() [3/4]

```
model.FloatParameter.FloatParameter (
    String name,
    String tag,
    Interval valueInterval )
```

Constructs a floating point number parameter with the provided name and tag, no description and an acceptance interval.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>valueInterval</i>	the interval in which the value is considered a valid entry

Definition at line 126 of file FloatParameter.java.

FloatParameter() [4/4]

```
model.FloatParameter.FloatParameter (
    String name,
    String tag,
```

```
String description,
Interval valueInterval )
```

Constructs a floating point number parameter with the provided name, tag, description and acceptance interval.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node
<i>valueInterval</i>	the interval in which the value is considered a valid entry

Definition at line 145 of file FloatParameter.java.

6.40.3 Member Function Documentation

isValid()

```
boolean model.FloatParameter.isValid ( )
```

Returns

true if the value string can be parsed to a decimal number within the parameter's acceptance interval.

See also

`java.lang.Float.valueOf(String s)`

Definition at line 157 of file FloatParameter.java.

setUpController()

```
NodeController model.FloatParameter.setUpController ( )
```

Builds a controller for this node.

Returns

the controller for this node.

See also

[ui.FieldParameterController](#)

Definition at line 182 of file FloatParameter.java.

The documentation for this class was generated from the following file:

- `src/model/FloatParameter.java`

6.41 visualization.fs2d.FS2DFile.Format Enum Reference

The different formats of files produced by FullSWOF_2D.

Public Attributes

- [GNUPLOT](#)
- [VTK](#)

6.41.1 Detailed Description

The different formats of files produced by FullSWOF_2D.

Definition at line 107 of file FS2DFile.java.

6.41.2 Member Data Documentation

GNUPLOT

`visualization.fs2d.FS2DFile.Format.GNUPLOT`

Definition at line 108 of file FS2DFile.java.

VTK

`visualization.fs2d.FS2DFile.Format.VTK`

Definition at line 108 of file FS2DFile.java.

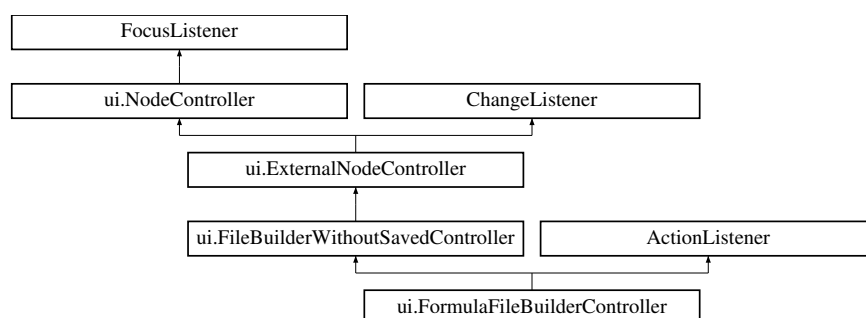
The documentation for this enum was generated from the following file:

- `src/visualization/fs2d/FS2DFile.java`

6.42 ui.FormulaFileBuilderController Class Reference

A controller for a file builder using parsed formulas.

Inheritance diagram for `ui.FormulaFileBuilderController`:



Public Member Functions

- [FormulaFileBuilderController](#) ([FormulaFileBuilderParameter](#) model)
- void [actionPerformed](#) (ActionEvent evt)

Package Functions

- void [highlightView](#) ()
- void [updateModel](#) ()
- void [updateView](#) ()
- void [setUpView](#) ()

Additional Inherited Members

6.42.1 Detailed Description

A controller for a file builder using parsed formulas.

Definition at line 88 of file FormulaFileBuilderController.java.

6.42.2 Constructor & Destructor Documentation

FormulaFileBuilderController()

```
ui.FormulaFileBuilderController.FormulaFileBuilderController (
    FormulaFileBuilderParameter model )
```

Constructs a controller for a file builder node.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 108 of file FormulaFileBuilderController.java.

6.42.3 Member Function Documentation

actionPerformed()

```
void ui.FormulaFileBuilderController.actionPerformed (
    ActionEvent evt )
```

Called when the user clicks on the 'visualize' button.

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 188 of file FormulaFileBuilderController.java.

highlightView()

```
void ui.FormulaFileBuilderController.highlightView ( ) [package]
```

Definition at line 113 of file FormulaFileBuilderController.java.

setUpView()

```
void ui.FormulaFileBuilderController.setUpView ( ) [package]
```

Definition at line 146 of file FormulaFileBuilderController.java.

updateModel()

```
void ui.FormulaFileBuilderController.updateModel ( ) [package]
```

Definition at line 121 of file FormulaFileBuilderController.java.

updateView()

```
void ui.FormulaFileBuilderController.updateView ( ) [package]
```

Definition at line 131 of file FormulaFileBuilderController.java.

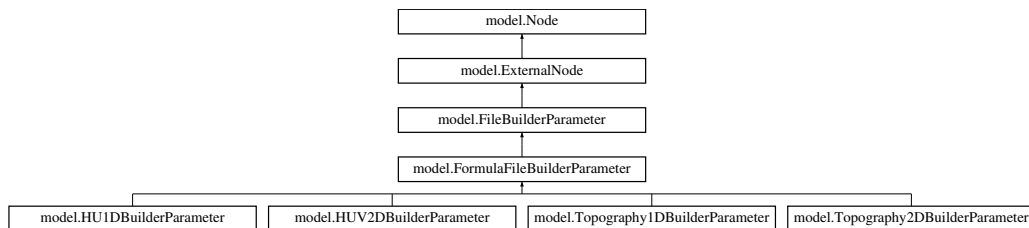
The documentation for this class was generated from the following file:

- [src/ui/FormulaFileBuilderController.java](#)

6.43 model.FormulaFileBuilderParameter Class Reference

A file builder parameter that uses a set of mathematical formulas to build a file.

Inheritance diagram for model.FormulaFileBuilderParameter:

**Public Member Functions**

- [FormulaFileBuilderParameter](#) (String [name](#), String [fileName](#))
- String [getFormula](#) (int i)
- void [setFormula](#) (int i, String formula)
- String [getFormulaLabel](#) (int i)
- [InputFileVisualizer](#) [getVisualizer](#) ()
- int [getNumberOfFormulas](#) ()
- boolean [isFormulaValid](#) (int i)
- abstract String [getFileContent](#) ()
- boolean [isValid](#) ()
- [NodeController](#) [setUpController](#) ()

Protected Member Functions

- double [parseFormula](#) (String formula)

Protected Attributes

- String [] [formulas](#)
The formulas used by the builder, stored as string.
- String [] [formulaLabels](#)
The name of the formulas, as displayed in the UI.
- [InputFileVisualizer](#) [visualizer](#)
The file visualizer used to get a quick visualization of the file produced.

Additional Inherited Members**6.43.1 Detailed Description**

A file builder parameter that uses a set of mathematical formulas to build a file.

The actual use of these formulas depend on the implementation

See also

[parser](#)

Definition at line 81 of file FormulaFileBuilderParameter.java.

6.43.2 Constructor & Destructor Documentation

FormulaFileBuilderParameter()

```
model.FormulaFileBuilderParameter.FormulaFileBuilderParameter (
    String name,
    String fileName )
```

Constructs a file builder with the provided name and file name.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file to be written

Definition at line 92 of file FormulaFileBuilderParameter.java.

6.43.3 Member Function Documentation

getFileContent()

```
abstract String model.FormulaFileBuilderParameter.getFileContent ( ) [abstract]
```

getFormula()

```
String model.FormulaFileBuilderParameter.getFormula (
    int i )
```

Return the formula at index i.

Parameters

<i>i</i>	the index of the formula
----------	--------------------------

Returns

a string representing a mathematical formula

Definition at line 120 of file FormulaFileBuilderParameter.java.

getFormulaLabel()

```
String model.FormulaFileBuilderParameter.getFormulaLabel (
    int i )
```

Parameters

<i>i</i>	the index of the formula
----------	--------------------------

Returns

the display name of the formula at index *i*.

Definition at line 146 of file FormulaFileBuilderParameter.java.

getNumberOfFormulas()

```
int model.FormulaFileBuilderParameter.getNumberOfFormulas ( )
```

Returns

the number of formulas used by this file builder.

Definition at line 164 of file FormulaFileBuilderParameter.java.

getVisualizer()

```
InputFileVisualizer model.FormulaFileBuilderParameter.getVisualizer ( )
```

Returns the file visualizer used to get a quick visualization of the file produced, which can be null.

Returns

a file visualizer or null.

Definition at line 157 of file FormulaFileBuilderParameter.java.

isFormulaValid()

```
boolean model.FormulaFileBuilderParameter.isFormulaValid (
    int i )
```

Parameters

<i>i</i>	the index of the formula
----------	--------------------------

Returns

true if the syntax of the formula at index *i* is valid.

Definition at line 174 of file FormulaFileBuilderParameter.java.

isValid()

```
boolean model.FormulaFileBuilderParameter.isValid ( )
```

Definition at line 182 of file FormulaFileBuilderParameter.java.

parseFormula()

```
double model.FormulaFileBuilderParameter.parseFormula (
    String formula ) [protected]
```

Returns the double value of a formula. Set the value of variables x, y and t in the parser's memory before using this method. If the formula is not valid, this method will return 0.0.

Parameters

<i>formula</i>	the string to be parsed as a formula
----------------	--------------------------------------

Returns

the double value of the formula.

Definition at line 237 of file FormulaFileBuilderParameter.java.

setFormula()

```
void model.FormulaFileBuilderParameter.setFormula (
    int i,
    String formula )
```

Sets the formula at index i.

Parameters

<i>i</i>	the index of the formula
<i>formula</i>	a string representing a mathematical formula.

Definition at line 133 of file FormulaFileBuilderParameter.java.

setUpController()

```
NodeController model.FormulaFileBuilderParameter.setUpController ( )
```

Definition at line 193 of file FormulaFileBuilderParameter.java.

6.43.4 Member Data Documentation**formulaLabels**

```
String [] model.FormulaFileBuilderParameter.formulaLabels [protected]
```

The name of the formulas, as displayed in the UI.

Definition at line 104 of file FormulaFileBuilderParameter.java.

formulas

```
String [] model.FormulaFileBuilderParameter.formulas [protected]
```

The formulas used by the builder, stored as string.

Definition at line 99 of file FormulaFileBuilderParameter.java.

visualizer

`InputFileVisualizer` `model.FormulaFileBuilderParameter.visualizer` [protected]

The file visualizer used to get a quick visualization of the file produced.

Can be null

Definition at line 110 of file `FormulaFileBuilderParameter.java`.

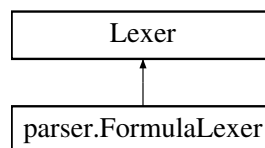
The documentation for this class was generated from the following file:

- `src/model/FormulaFileBuilderParameter.java`

6.44 parser.FormulaLexer Class Reference

Transforms the character stream into a series of tokens.

Inheritance diagram for `parser.FormulaLexer`:

**Classes**

- class [DFA12](#)
- class [DFA8](#)

Public Member Functions

- void [reportError](#) (RecognitionException e)
- void [recover](#) (RecognitionException e)
- Lexer [] [getDelegates](#) ()
- [FormulaLexer](#) ()
- [FormulaLexer](#) (CharStream input)
- [FormulaLexer](#) (CharStream input, RecognizerSharedState state)
- String [getGrammarFileName](#) ()
- final void [mT__8](#) () throws RecognitionException
- final void [mT__9](#) () throws RecognitionException
- final void [mT__10](#) () throws RecognitionException
- final void [mT__11](#) () throws RecognitionException
- final void [mT__12](#) () throws RecognitionException
- final void [mT__13](#) () throws RecognitionException
- final void [mT__14](#) () throws RecognitionException
- final void [mT__15](#) () throws RecognitionException
- final void [mT__16](#) () throws RecognitionException
- final void [mT__17](#) () throws RecognitionException
- final void [mT__18](#) () throws RecognitionException
- final void [mT__19](#) () throws RecognitionException
- final void [mT__20](#) () throws RecognitionException
- final void [mT__21](#) () throws RecognitionException
- final void [mT__22](#) () throws RecognitionException
- final void [mT__23](#) () throws RecognitionException
- final void [mT__24](#) () throws RecognitionException

- final void `mT__25` () throws `RecognitionException`
- final void `mT__26` () throws `RecognitionException`
- final void `mT__27` () throws `RecognitionException`
- final void `mT__28` () throws `RecognitionException`
- final void `mT__29` () throws `RecognitionException`
- final void `mT__30` () throws `RecognitionException`
- final void `mT__31` () throws `RecognitionException`
- final void `mT__32` () throws `RecognitionException`
- final void `mT__33` () throws `RecognitionException`
- final void `mT__34` () throws `RecognitionException`
- final void `mT__35` () throws `RecognitionException`
- final void `mT__36` () throws `RecognitionException`
- final void `mT__37` () throws `RecognitionException`
- final void `mT__38` () throws `RecognitionException`
- final void `mT__39` () throws `RecognitionException`
- final void `mT__40` () throws `RecognitionException`
- final void `mT__41` () throws `RecognitionException`
- final void `mT__42` () throws `RecognitionException`
- final void `mID` () throws `RecognitionException`
- final void `mFLOAT` () throws `RecognitionException`
- final void `mEXPONENT` () throws `RecognitionException`
- final void `mNEWLINE` () throws `RecognitionException`
- void `mTokens` () throws `RecognitionException`

Static Public Attributes

- static final int `EOF`
- static final int `T__8`
- static final int `T__9`
- static final int `T__10`
- static final int `T__11`
- static final int `T__12`
- static final int `T__13`
- static final int `T__14`
- static final int `T__15`
- static final int `T__16`
- static final int `T__17`
- static final int `T__18`
- static final int `T__19`
- static final int `T__20`
- static final int `T__21`
- static final int `T__22`
- static final int `T__23`
- static final int `T__24`
- static final int `T__25`
- static final int `T__26`
- static final int `T__27`
- static final int `T__28`
- static final int `T__29`

- static final int [T__30](#)
- static final int [T__31](#)
- static final int [T__32](#)
- static final int [T__33](#)
- static final int [T__34](#)
- static final int [T__35](#)
- static final int [T__36](#)
- static final int [T__37](#)
- static final int [T__38](#)
- static final int [T__39](#)
- static final int [T__40](#)
- static final int [T__41](#)
- static final int [T__42](#)
- static final int [EXPONENT](#)
- static final int [FLOAT](#)
- static final int [ID](#)
- static final int [NEWLINE](#)

Protected Attributes

- [DFA8 dfa8](#)
- [DFA12 dfa12](#)

Static Package Functions

- [\[static initializer\]](#)
- [\[static initializer\]](#)

Static Package Attributes

- static final String [DFA8_eotS](#)
- static final String [DFA8_eofS](#)
- static final String [DFA8_minS](#)
- static final String [DFA8_maxS](#)
- static final String [DFA8_acceptS](#)
- static final String [DFA8_specialS](#)
- static final String [] [DFA8_transitionS](#)
- static final short [] [DFA8_eot](#)
- static final short [] [DFA8_eof](#)
- static final char [] [DFA8_min](#)
- static final char [] [DFA8_max](#)
- static final short [] [DFA8_accept](#)
- static final short [] [DFA8_special](#)
- static final short [][] [DFA8_transition](#)
- static final String [DFA12_eotS](#)
- static final String [DFA12_eofS](#)
- static final String [DFA12_minS](#)
- static final String [DFA12_maxS](#)
- static final String [DFA12_acceptS](#)
- static final String [DFA12_specialS](#)

- static final String [] [DFA12_transitionS](#)
- static final short [] [DFA12_eot](#)
- static final short [] [DFA12_eof](#)
- static final char [] [DFA12_min](#)
- static final char [] [DFA12_max](#)
- static final short [] [DFA12_accept](#)
- static final short [] [DFA12_special](#)
- static final short [][] [DFA12_transition](#)

6.44.1 Detailed Description

Transforms the character stream into a series of tokens.

Definition at line 71 of file FormulaLexer.java.

6.44.2 Constructor & Destructor Documentation

FormulaLexer() [1/3]

```
parser.FormulaLexer.FormulaLexer ( )
```

Definition at line 129 of file FormulaLexer.java.

FormulaLexer() [2/3]

```
parser.FormulaLexer.FormulaLexer (
    CharStream input )
```

Definition at line 132 of file FormulaLexer.java.

FormulaLexer() [3/3]

```
parser.FormulaLexer.FormulaLexer (
    CharStream input,
    RecognizerSharedState state )
```

Definition at line 136 of file FormulaLexer.java.

6.44.3 Member Function Documentation

[static initializer]() [1/2]

```
parser.FormulaLexer.[static initializer] ( ) [static], [package]
```

[static initializer]() [2/2]

```
parser.FormulaLexer.[static initializer] ( ) [static], [package]
```

getDelegates()

```
Lexer [] parser.FormulaLexer.getDelegates ( )
```

Definition at line 125 of file FormulaLexer.java.

getGrammarFileName()

```
String parser.FormulaLexer.getGrammarFileName ( )
```

Definition at line 140 of file FormulaLexer.java.

mEXPONENT()

```
final void parser.FormulaLexer.mEXPONENT ( ) throws RecognitionException
```

Definition at line 1061 of file FormulaLexer.java.

mFLOAT()

```
final void parser.FormulaLexer.mFLOAT ( ) throws RecognitionException
```

Definition at line 834 of file FormulaLexer.java.

mID()

```
final void parser.FormulaLexer.mID ( ) throws RecognitionException
```

Definition at line 808 of file FormulaLexer.java.

mNEWLINE()

```
final void parser.FormulaLexer.mNEWLINE ( ) throws RecognitionException
```

Definition at line 1145 of file FormulaLexer.java.

mT__10()

```
final void parser.FormulaLexer.mT__10 ( ) throws RecognitionException
```

Definition at line 182 of file FormulaLexer.java.

mT__11()

```
final void parser.FormulaLexer.mT__11 ( ) throws RecognitionException
```

Definition at line 201 of file FormulaLexer.java.

mT__12()

```
final void parser.FormulaLexer.mT__12 ( ) throws RecognitionException
```

Definition at line 220 of file FormulaLexer.java.

mT__13()

`final void parser.FormulaLexer.mT__13 () throws RecognitionException`
Definition at line 239 of file FormulaLexer.java.

mT__14()

`final void parser.FormulaLexer.mT__14 () throws RecognitionException`
Definition at line 258 of file FormulaLexer.java.

mT__15()

`final void parser.FormulaLexer.mT__15 () throws RecognitionException`
Definition at line 277 of file FormulaLexer.java.

mT__16()

`final void parser.FormulaLexer.mT__16 () throws RecognitionException`
Definition at line 296 of file FormulaLexer.java.

mT__17()

`final void parser.FormulaLexer.mT__17 () throws RecognitionException`
Definition at line 315 of file FormulaLexer.java.

mT__18()

`final void parser.FormulaLexer.mT__18 () throws RecognitionException`
Definition at line 334 of file FormulaLexer.java.

mT__19()

`final void parser.FormulaLexer.mT__19 () throws RecognitionException`
Definition at line 353 of file FormulaLexer.java.

mT__20()

`final void parser.FormulaLexer.mT__20 () throws RecognitionException`
Definition at line 372 of file FormulaLexer.java.

mT__21()

`final void parser.FormulaLexer.mT__21 () throws RecognitionException`
Definition at line 391 of file FormulaLexer.java.

mT__22()

`final void parser.FormulaLexer.mT__22 () throws RecognitionException`
Definition at line 409 of file FormulaLexer.java.

mT__23()

`final void parser.FormulaLexer.mT__23 () throws RecognitionException`
Definition at line 428 of file FormulaLexer.java.

mT__24()

`final void parser.FormulaLexer.mT__24 () throws RecognitionException`
Definition at line 447 of file FormulaLexer.java.

mT__25()

`final void parser.FormulaLexer.mT__25 () throws RecognitionException`
Definition at line 466 of file FormulaLexer.java.

mT__26()

`final void parser.FormulaLexer.mT__26 () throws RecognitionException`
Definition at line 485 of file FormulaLexer.java.

mT__27()

`final void parser.FormulaLexer.mT__27 () throws RecognitionException`
Definition at line 504 of file FormulaLexer.java.

mT__28()

`final void parser.FormulaLexer.mT__28 () throws RecognitionException`
Definition at line 523 of file FormulaLexer.java.

mT__29()

`final void parser.FormulaLexer.mT__29 () throws RecognitionException`
Definition at line 542 of file FormulaLexer.java.

mT__30()

`final void parser.FormulaLexer.mT__30 () throws RecognitionException`
Definition at line 561 of file FormulaLexer.java.

mT__31()

`final void parser.FormulaLexer.mT__31 () throws RecognitionException`
Definition at line 580 of file FormulaLexer.java.

mT__32()

`final void parser.FormulaLexer.mT__32 () throws RecognitionException`
Definition at line 599 of file FormulaLexer.java.

mT__33()

`final void parser.FormulaLexer.mT__33 () throws RecognitionException`
Definition at line 618 of file FormulaLexer.java.

mT__34()

`final void parser.FormulaLexer.mT__34 () throws RecognitionException`
Definition at line 637 of file FormulaLexer.java.

mT__35()

`final void parser.FormulaLexer.mT__35 () throws RecognitionException`
Definition at line 656 of file FormulaLexer.java.

mT__36()

`final void parser.FormulaLexer.mT__36 () throws RecognitionException`
Definition at line 675 of file FormulaLexer.java.

mT__37()

`final void parser.FormulaLexer.mT__37 () throws RecognitionException`
Definition at line 694 of file FormulaLexer.java.

mT__38()

`final void parser.FormulaLexer.mT__38 () throws RecognitionException`
Definition at line 713 of file FormulaLexer.java.

mT__39()

`final void parser.FormulaLexer.mT__39 () throws RecognitionException`
Definition at line 732 of file FormulaLexer.java.

mT__40()

`final void parser.FormulaLexer.mT__40 () throws RecognitionException`
Definition at line 751 of file FormulaLexer.java.

mT__41()

`final void parser.FormulaLexer.mT__41 () throws RecognitionException`
Definition at line 770 of file FormulaLexer.java.

mT__42()

`final void parser.FormulaLexer.mT__42 () throws RecognitionException`
Definition at line 789 of file FormulaLexer.java.

mT__8()

`final void parser.FormulaLexer.mT__8 () throws RecognitionException`
Definition at line 144 of file FormulaLexer.java.

mT__9()

`final void parser.FormulaLexer.mT__9 () throws RecognitionException`
Definition at line 163 of file FormulaLexer.java.

mTokens()

`void parser.FormulaLexer.mTokens () throws RecognitionException`
Definition at line 1180 of file FormulaLexer.java.

recover()

`void parser.FormulaLexer.recover (`
 `RecognitionException e)`
Definition at line 119 of file FormulaLexer.java.

reportError()

`void parser.FormulaLexer.reportError (`
 `RecognitionException e)`
Definition at line 114 of file FormulaLexer.java.

6.44.4 Member Data Documentation

dfa12

`DFA12` `parser.FormulaLexer.dfa12` [protected]
Definition at line 1469 of file `FormulaLexer.java`.

DFA12_accept

`final short []` `parser.FormulaLexer.DFA12_accept` [static], [package]
Definition at line 1554 of file `FormulaLexer.java`.

DFA12_acceptS

`final String` `parser.FormulaLexer.DFA12_acceptS` [static], [package]
Definition at line 1525 of file `FormulaLexer.java`.

DFA12_eof

`final short []` `parser.FormulaLexer.DFA12_eof` [static], [package]
Definition at line 1549 of file `FormulaLexer.java`.

DFA12_eofS

`final String` `parser.FormulaLexer.DFA12_eofS` [static], [package]
Definition at line 1517 of file `FormulaLexer.java`.

DFA12_eot

`final short []` `parser.FormulaLexer.DFA12_eot` [static], [package]
Definition at line 1548 of file `FormulaLexer.java`.

DFA12_eotS

`final String` `parser.FormulaLexer.DFA12_eotS` [static], [package]
Definition at line 1516 of file `FormulaLexer.java`.

DFA12_max

`final char []` `parser.FormulaLexer.DFA12_max` [static], [package]
Definition at line 1552 of file `FormulaLexer.java`.

DFA12_maxS

`final String` `parser.FormulaLexer.DFA12_maxS` [static], [package]
Definition at line 1521 of file `FormulaLexer.java`.

DFA12_min

```
final char [] parser.FormulaLexer.DFA12_min [static], [package]
```

Definition at line 1550 of file FormulaLexer.java.

DFA12_minS

```
final String parser.FormulaLexer.DFA12_minS [static], [package]
```

Definition at line 1518 of file FormulaLexer.java.

DFA12_special

```
final short [] parser.FormulaLexer.DFA12_special [static], [package]
```

Definition at line 1555 of file FormulaLexer.java.

DFA12_specialS

```
final String parser.FormulaLexer.DFA12_specialS [static], [package]
```

Definition at line 1530 of file FormulaLexer.java.

DFA12_transition

```
final short [][] parser.FormulaLexer.DFA12_transition [static], [package]
```

Definition at line 1557 of file FormulaLexer.java.

DFA12_transitionS

```
final String [] parser.FormulaLexer.DFA12_transitionS [static], [package]
```

Definition at line 1531 of file FormulaLexer.java.

dfa8

```
DFA8 parser.FormulaLexer.dfa8 [protected]
```

Definition at line 1468 of file FormulaLexer.java.

DFA8_accept

```
final short [] parser.FormulaLexer.DFA8_accept [static], [package]
```

Definition at line 1485 of file FormulaLexer.java.

DFA8_acceptS

```
final String parser.FormulaLexer.DFA8_acceptS [static], [package]
```

Definition at line 1474 of file FormulaLexer.java.

DFA8_eof

```
final short [] parser.FormulaLexer.DFA8_eof [static], [package]
```

Definition at line 1480 of file FormulaLexer.java.

DFA8_eofS

```
final String parser.FormulaLexer.DFA8_eofS [static], [package]
```

Definition at line 1471 of file FormulaLexer.java.

DFA8_eot

```
final short [] parser.FormulaLexer.DFA8_eot [static], [package]
```

Definition at line 1479 of file FormulaLexer.java.

DFA8_eotS

```
final String parser.FormulaLexer.DFA8_eotS [static], [package]
```

Definition at line 1470 of file FormulaLexer.java.

DFA8_max

```
final char [] parser.FormulaLexer.DFA8_max [static], [package]
```

Definition at line 1483 of file FormulaLexer.java.

DFA8_maxS

```
final String parser.FormulaLexer.DFA8_maxS [static], [package]
```

Definition at line 1473 of file FormulaLexer.java.

DFA8_min

```
final char [] parser.FormulaLexer.DFA8_min [static], [package]
```

Definition at line 1481 of file FormulaLexer.java.

DFA8_minS

```
final String parser.FormulaLexer.DFA8_minS [static], [package]
```

Definition at line 1472 of file FormulaLexer.java.

DFA8_special

```
final short [] parser.FormulaLexer.DFA8_special [static], [package]
```

Definition at line 1486 of file FormulaLexer.java.

DFA8_specialS

```
final String parser.FormulaLexer.DFA8_specialS [static], [package]
```

Definition at line 1475 of file FormulaLexer.java.

DFA8_transition

```
final short [][] parser.FormulaLexer.DFA8_transition [static], [package]
```

Definition at line 1487 of file FormulaLexer.java.

DFA8_transitionS

```
final String [] parser.FormulaLexer.DFA8_transitionS [static], [package]
```

Definition at line 1476 of file FormulaLexer.java.

EOF

```
final int parser.FormulaLexer.EOF [static]
```

Definition at line 72 of file FormulaLexer.java.

EXPONENT

```
final int parser.FormulaLexer.EXPONENT [static]
```

Definition at line 108 of file FormulaLexer.java.

FLOAT

```
final int parser.FormulaLexer.FLOAT [static]
```

Definition at line 109 of file FormulaLexer.java.

ID

```
final int parser.FormulaLexer.ID [static]
```

Definition at line 110 of file FormulaLexer.java.

NEWLINE

```
final int parser.FormulaLexer.NEWLINE [static]
```

Definition at line 111 of file FormulaLexer.java.

T__10

```
final int parser.FormulaLexer.T__10 [static]
```

Definition at line 75 of file FormulaLexer.java.

T__11

```
final int parser.FormulaLexer.T__11 [static]
```

Definition at line 76 of file FormulaLexer.java.

T__12

```
final int parser.FormulaLexer.T__12 [static]
```

Definition at line 77 of file FormulaLexer.java.

T__13

```
final int parser.FormulaLexer.T__13 [static]
```

Definition at line 78 of file FormulaLexer.java.

T__14

```
final int parser.FormulaLexer.T__14 [static]
```

Definition at line 79 of file FormulaLexer.java.

T__15

```
final int parser.FormulaLexer.T__15 [static]
```

Definition at line 80 of file FormulaLexer.java.

T__16

```
final int parser.FormulaLexer.T__16 [static]
```

Definition at line 81 of file FormulaLexer.java.

T__17

```
final int parser.FormulaLexer.T__17 [static]
```

Definition at line 82 of file FormulaLexer.java.

T__18

```
final int parser.FormulaLexer.T__18 [static]
```

Definition at line 83 of file FormulaLexer.java.

T__19

```
final int parser.FormulaLexer.T__19 [static]
```

Definition at line 84 of file FormulaLexer.java.

T__20

```
final int parser.FormulaLexer.T__20 [static]
```

Definition at line 85 of file FormulaLexer.java.

T__21

```
final int parser.FormulaLexer.T__21 [static]
```

Definition at line 86 of file FormulaLexer.java.

T__22

```
final int parser.FormulaLexer.T__22 [static]
```

Definition at line 87 of file FormulaLexer.java.

T__23

```
final int parser.FormulaLexer.T__23 [static]
```

Definition at line 88 of file FormulaLexer.java.

T__24

```
final int parser.FormulaLexer.T__24 [static]
```

Definition at line 89 of file FormulaLexer.java.

T__25

```
final int parser.FormulaLexer.T__25 [static]
```

Definition at line 90 of file FormulaLexer.java.

T__26

```
final int parser.FormulaLexer.T__26 [static]
```

Definition at line 91 of file FormulaLexer.java.

T__27

```
final int parser.FormulaLexer.T__27 [static]
```

Definition at line 92 of file FormulaLexer.java.

T__28

```
final int parser.FormulaLexer.T__28 [static]
```

Definition at line 93 of file FormulaLexer.java.

T__29

```
final int parser.FormulaLexer.T__29 [static]
```

Definition at line 94 of file FormulaLexer.java.

T__30

```
final int parser.FormulaLexer.T__30 [static]
```

Definition at line 95 of file FormulaLexer.java.

T__31

```
final int parser.FormulaLexer.T__31 [static]
```

Definition at line 96 of file FormulaLexer.java.

T__32

```
final int parser.FormulaLexer.T__32 [static]
```

Definition at line 97 of file FormulaLexer.java.

T__33

```
final int parser.FormulaLexer.T__33 [static]
```

Definition at line 98 of file FormulaLexer.java.

T__34

```
final int parser.FormulaLexer.T__34 [static]
```

Definition at line 99 of file FormulaLexer.java.

T__35

```
final int parser.FormulaLexer.T__35 [static]
```

Definition at line 100 of file FormulaLexer.java.

T__36

```
final int parser.FormulaLexer.T__36 [static]
```

Definition at line 101 of file FormulaLexer.java.

T__37

```
final int parser.FormulaLexer.T__37 [static]
```

Definition at line 102 of file FormulaLexer.java.

T__38

```
final int parser.FormulaLexer.T__38 [static]
```

Definition at line 103 of file FormulaLexer.java.

T__39

```
final int parser.FormulaLexer.T__39 [static]
```

Definition at line 104 of file FormulaLexer.java.

T__40

```
final int parser.FormulaLexer.T__40 [static]
```

Definition at line 105 of file FormulaLexer.java.

T__41

```
final int parser.FormulaLexer.T__41 [static]
```

Definition at line 106 of file FormulaLexer.java.

T__42

```
final int parser.FormulaLexer.T__42 [static]
```

Definition at line 107 of file FormulaLexer.java.

T__8

```
final int parser.FormulaLexer.T__8 [static]
```

Definition at line 73 of file FormulaLexer.java.

T__9

```
final int parser.FormulaLexer.T__9 [static]
```

Definition at line 74 of file FormulaLexer.java.

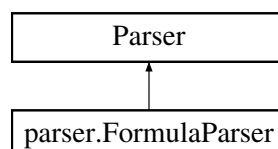
The documentation for this class was generated from the following file:

- src/parser/[FormulaLexer.java](#)

6.45 parser.FormulaParser Class Reference

Walks through the tokens to form mathematical sentences in the grammar.

Inheritance diagram for parser.FormulaParser:



Public Member Functions

- Parser [] [getDelegates](#) ()
- [FormulaParser](#) (TokenStream input)
- [FormulaParser](#) (TokenStream input, RecognizerSharedState state)
- String [] [getTokenNames](#) ()
- Object [recoverFromMismatchedSet](#) (IntStream input, RecognitionException e, BitSet follow) throws RecognitionException
- final double [formula](#) () throws RecognitionException
- final double [expr](#) () throws RecognitionException
- final double [multExpr](#) () throws RecognitionException
- final double [function](#) () throws RecognitionException
- final double [signedAtom](#) () throws RecognitionException
- final double [atom](#) () throws RecognitionException

Static Public Member Functions

- static HashMap< String, Double > [getMemory](#) ()

Static Public Attributes

- static final String [] [tokenNames](#)
- static final int [EOF](#)
- static final int [T__8](#)
- static final int [T__9](#)
- static final int [T__10](#)
- static final int [T__11](#)
- static final int [T__12](#)
- static final int [T__13](#)
- static final int [T__14](#)
- static final int [T__15](#)
- static final int [T__16](#)
- static final int [T__17](#)
- static final int [T__18](#)
- static final int [T__19](#)
- static final int [T__20](#)
- static final int [T__21](#)
- static final int [T__22](#)
- static final int [T__23](#)
- static final int [T__24](#)
- static final int [T__25](#)
- static final int [T__26](#)
- static final int [T__27](#)
- static final int [T__28](#)
- static final int [T__29](#)
- static final int [T__30](#)
- static final int [T__31](#)
- static final int [T__32](#)
- static final int [T__33](#)
- static final int [T__34](#)

- static final int [T__35](#)
- static final int [T__36](#)
- static final int [T__37](#)
- static final int [T__38](#)
- static final int [T__39](#)
- static final int [T__40](#)
- static final int [T__41](#)
- static final int [T__42](#)
- static final int [EXPONENT](#)
- static final int [FLOAT](#)
- static final int [ID](#)
- static final int [NEWLINE](#)
- static final BitSet [FOLLOW_expr_in_formula47](#)
- static final BitSet [FOLLOW_NEWLINE_in_formula49](#)
- static final BitSet [FOLLOW_NEWLINE_in_formula56](#)
- static final BitSet [FOLLOW_multExpr_in_expr73](#)
- static final BitSet [FOLLOW_11_in_expr80](#)
- static final BitSet [FOLLOW_multExpr_in_expr84](#)
- static final BitSet [FOLLOW_14_in_expr91](#)
- static final BitSet [FOLLOW_multExpr_in_expr95](#)
- static final BitSet [FOLLOW_signedAtom_in_multExpr120](#)
- static final BitSet [FOLLOW_10_in_multExpr128](#)
- static final BitSet [FOLLOW_signedAtom_in_multExpr132](#)
- static final BitSet [FOLLOW_15_in_multExpr139](#)
- static final BitSet [FOLLOW_signedAtom_in_multExpr143](#)
- static final BitSet [FOLLOW_16_in_function165](#)
- static final BitSet [FOLLOW_expr_in_function168](#)
- static final BitSet [FOLLOW_9_in_function169](#)
- static final BitSet [FOLLOW_17_in_function176](#)
- static final BitSet [FOLLOW_expr_in_function179](#)
- static final BitSet [FOLLOW_9_in_function180](#)
- static final BitSet [FOLLOW_18_in_function187](#)
- static final BitSet [FOLLOW_expr_in_function190](#)
- static final BitSet [FOLLOW_9_in_function191](#)
- static final BitSet [FOLLOW_19_in_function198](#)
- static final BitSet [FOLLOW_expr_in_function201](#)
- static final BitSet [FOLLOW_9_in_function202](#)
- static final BitSet [FOLLOW_20_in_function209](#)
- static final BitSet [FOLLOW_expr_in_function212](#)
- static final BitSet [FOLLOW_13_in_function213](#)
- static final BitSet [FOLLOW_expr_in_function216](#)
- static final BitSet [FOLLOW_9_in_function217](#)
- static final BitSet [FOLLOW_21_in_function224](#)
- static final BitSet [FOLLOW_expr_in_function227](#)
- static final BitSet [FOLLOW_9_in_function228](#)
- static final BitSet [FOLLOW_22_in_function235](#)
- static final BitSet [FOLLOW_expr_in_function238](#)
- static final BitSet [FOLLOW_9_in_function239](#)
- static final BitSet [FOLLOW_23_in_function246](#)

- static final BitSet [FOLLOW_expr_in_function249](#)
- static final BitSet [FOLLOW_9_in_function250](#)
- static final BitSet [FOLLOW_24_in_function257](#)
- static final BitSet [FOLLOW_expr_in_function260](#)
- static final BitSet [FOLLOW_9_in_function261](#)
- static final BitSet [FOLLOW_25_in_function268](#)
- static final BitSet [FOLLOW_expr_in_function271](#)
- static final BitSet [FOLLOW_9_in_function272](#)
- static final BitSet [FOLLOW_26_in_function279](#)
- static final BitSet [FOLLOW_expr_in_function282](#)
- static final BitSet [FOLLOW_9_in_function283](#)
- static final BitSet [FOLLOW_27_in_function290](#)
- static final BitSet [FOLLOW_expr_in_function293](#)
- static final BitSet [FOLLOW_9_in_function294](#)
- static final BitSet [FOLLOW_28_in_function301](#)
- static final BitSet [FOLLOW_expr_in_function304](#)
- static final BitSet [FOLLOW_12_in_function305](#)
- static final BitSet [FOLLOW_expr_in_function308](#)
- static final BitSet [FOLLOW_9_in_function309](#)
- static final BitSet [FOLLOW_29_in_function316](#)
- static final BitSet [FOLLOW_expr_in_function319](#)
- static final BitSet [FOLLOW_9_in_function320](#)
- static final BitSet [FOLLOW_30_in_function327](#)
- static final BitSet [FOLLOW_expr_in_function330](#)
- static final BitSet [FOLLOW_9_in_function331](#)
- static final BitSet [FOLLOW_31_in_function338](#)
- static final BitSet [FOLLOW_expr_in_function341](#)
- static final BitSet [FOLLOW_9_in_function342](#)
- static final BitSet [FOLLOW_32_in_function349](#)
- static final BitSet [FOLLOW_expr_in_function352](#)
- static final BitSet [FOLLOW_13_in_function353](#)
- static final BitSet [FOLLOW_expr_in_function356](#)
- static final BitSet [FOLLOW_9_in_function357](#)
- static final BitSet [FOLLOW_33_in_function364](#)
- static final BitSet [FOLLOW_expr_in_function367](#)
- static final BitSet [FOLLOW_13_in_function368](#)
- static final BitSet [FOLLOW_expr_in_function371](#)
- static final BitSet [FOLLOW_9_in_function372](#)
- static final BitSet [FOLLOW_34_in_function379](#)
- static final BitSet [FOLLOW_expr_in_function382](#)
- static final BitSet [FOLLOW_13_in_function383](#)
- static final BitSet [FOLLOW_expr_in_function386](#)
- static final BitSet [FOLLOW_9_in_function387](#)
- static final BitSet [FOLLOW_35_in_function394](#)
- static final BitSet [FOLLOW_36_in_function401](#)
- static final BitSet [FOLLOW_expr_in_function404](#)
- static final BitSet [FOLLOW_9_in_function405](#)
- static final BitSet [FOLLOW_37_in_function412](#)
- static final BitSet [FOLLOW_expr_in_function415](#)

- static final BitSet [FOLLOW_9_in_function416](#)
- static final BitSet [FOLLOW_38_in_function423](#)
- static final BitSet [FOLLOW_expr_in_function426](#)
- static final BitSet [FOLLOW_9_in_function427](#)
- static final BitSet [FOLLOW_39_in_function434](#)
- static final BitSet [FOLLOW_expr_in_function437](#)
- static final BitSet [FOLLOW_9_in_function438](#)
- static final BitSet [FOLLOW_40_in_function445](#)
- static final BitSet [FOLLOW_expr_in_function448](#)
- static final BitSet [FOLLOW_9_in_function449](#)
- static final BitSet [FOLLOW_41_in_function456](#)
- static final BitSet [FOLLOW_expr_in_function459](#)
- static final BitSet [FOLLOW_9_in_function460](#)
- static final BitSet [FOLLOW_42_in_function467](#)
- static final BitSet [FOLLOW_expr_in_function470](#)
- static final BitSet [FOLLOW_9_in_function471](#)
- static final BitSet [FOLLOW_11_in_signedAtom489](#)
- static final BitSet [FOLLOW_atom_in_signedAtom494](#)
- static final BitSet [FOLLOW_14_in_signedAtom501](#)
- static final BitSet [FOLLOW_atom_in_signedAtom505](#)
- static final BitSet [FOLLOW_FLOAT_in_atom522](#)
- static final BitSet [FOLLOW_ID_in_atom529](#)
- static final BitSet [FOLLOW_8_in_atom537](#)
- static final BitSet [FOLLOW_expr_in_atom539](#)
- static final BitSet [FOLLOW_9_in_atom541](#)
- static final BitSet [FOLLOW_function_in_atom548](#)

Protected Member Functions

- Object [recoverFromMismatchedToken](#) (IntStream input, int ttype, BitSet follow) throws Recognition↔Exception

6.45.1 Detailed Description

Walks though the tokens to form mathematical sentences in the grammar.
Definition at line 73 of file FormulaParser.java.

6.45.2 Constructor & Destructor Documentation

FormulaParser() [1/2]

```
parser.FormulaParser.FormulaParser (
    TokenStream input )
```

Definition at line 131 of file FormulaParser.java.

FormulaParser() [2/2]

```
parser.FormulaParser.FormulaParser (
    TokenStream input,
    RecognizerSharedState state )
Definition at line 135 of file FormulaParser.java.
```

6.45.3 Member Function Documentation**atom()**

```
final double parser.FormulaParser.atom ( ) throws RecognitionException
```

Returns

value double (FLOAT | ID | '(' expr ')' | function)

Definition at line 1106 of file FormulaParser.java.

expr()

```
final double parser.FormulaParser.expr ( ) throws RecognitionException
```

Returns

value double e= multExpr ('+' e= multExpr | '-' e= multExpr)*

Definition at line 226 of file FormulaParser.java.

formula()

```
final double parser.FormulaParser.formula ( ) throws RecognitionException
```

Returns

value double (e= expr NEWLINE | NEWLINE)

Definition at line 161 of file FormulaParser.java.

function()

```
final double parser.FormulaParser.function ( ) throws RecognitionException
```

Returns

value double ('abs(' e= expr ')' | 'acos(' e= expr ')' | 'asin(' e= expr ')' | 'atan(' e= expr ')' | 'atan2(' e1= expr
, ' e2= expr ')' | 'cbrt(' e= expr ')' | 'ceil(' e= expr ')' | 'cos(' e= expr ')' | 'cosh(' e= expr ')' | 'exp(' e= expr ')' |
'expm1(' e= expr ')' | 'floor(' e= expr ')' | 'hypot(' e1= expr ', ' e2= expr ')' | 'log(' e= expr ')' | 'log10(' e= expr
)' | 'log1p(' e= expr ')' | 'max(' e1= expr ', ' e2= expr ')' | 'min(' e1= expr ', ' e2= expr ')' | 'pow(' e1= expr ', ' e2=
expr ')' | 'random()' | 'rint(' e= expr ')' | 'signum(' e= expr ')' | 'sin(' e= expr ')' | 'sinh(' e= expr ')' | 'sqrt(' e=
expr ')' | 'tan(' e= expr ')' | 'tanh(' e= expr ')')

Definition at line 390 of file FormulaParser.java.

getDelegates()

```
Parser [] parser.FormulaParser.getDelegates ( )
```

Definition at line 125 of file FormulaParser.java.

getMemory()

```
static HashMap<String, Double> parser.FormulaParser.getMemory ( ) [static]
```

Definition at line 145 of file FormulaParser.java.

getTokenNames()

```
String [] parser.FormulaParser.getTokenNames ( )
```

Definition at line 139 of file FormulaParser.java.

multExpr()

```
final double parser.FormulaParser.multExpr ( ) throws RecognitionException
```

Returns

value double e= signedAtom ('*' e= signedAtom | '/' e= signedAtom)*

Definition at line 307 of file FormulaParser.java.

recoverFromMismatchedSet()

```
Object parser.FormulaParser.recoverFromMismatchedSet (
    IntStream input,
    RecognitionException e,
    BitSet follow ) throws RecognitionException
```

Definition at line 156 of file FormulaParser.java.

recoverFromMismatchedToken()

```
Object parser.FormulaParser.recoverFromMismatchedToken (
    IntStream input,
    int ttype,
    BitSet follow ) throws RecognitionException [protected]
```

Definition at line 150 of file FormulaParser.java.

signedAtom()

```
final double parser.FormulaParser.signedAtom ( ) throws RecognitionException
```

Returns

value double (('+')? e= atom | '-' e= atom)

Definition at line 1019 of file FormulaParser.java.

6.45.4 Member Data Documentation

EOF

```
final int parser.FormulaParser.EOF [static]
```

Definition at line 83 of file FormulaParser.java.

EXPONENT

```
final int parser.FormulaParser.EXPONENT [static]
```

Definition at line 119 of file FormulaParser.java.

FLOAT

```
final int parser.FormulaParser.FLOAT [static]
```

Definition at line 120 of file FormulaParser.java.

FOLLOW_10_in_multExpr128

```
final BitSet parser.FormulaParser.FOLLOW_10_in_multExpr128 [static]
```

Definition at line 1259 of file FormulaParser.java.

FOLLOW_11_in_expr80

```
final BitSet parser.FormulaParser.FOLLOW_11_in_expr80 [static]
```

Definition at line 1249 of file FormulaParser.java.

FOLLOW_11_in_signedAtom489

```
final BitSet parser.FormulaParser.FOLLOW_11_in_signedAtom489 [static]
```

Definition at line 1445 of file FormulaParser.java.

FOLLOW_12_in_function305

```
final BitSet parser.FormulaParser.FOLLOW_12_in_function305 [static]
```

Definition at line 1347 of file FormulaParser.java.

FOLLOW_13_in_function213

```
final BitSet parser.FormulaParser.FOLLOW_13_in_function213 [static]
```

Definition at line 1295 of file FormulaParser.java.

FOLLOW_13_in_function353

```
final BitSet parser.FormulaParser.FOLLOW_13_in_function353 [static]
```

Definition at line 1375 of file FormulaParser.java.

FOLLOW_13_in_function368

```
final BitSet parser.FormulaParser.FOLLOW_13_in_function368 [static]
```

Definition at line 1385 of file FormulaParser.java.

FOLLOW_13_in_function383

```
final BitSet parser.FormulaParser.FOLLOW_13_in_function383 [static]
```

Definition at line 1395 of file FormulaParser.java.

FOLLOW_14_in_expr91

```
final BitSet parser.FormulaParser.FOLLOW_14_in_expr91 [static]
```

Definition at line 1253 of file FormulaParser.java.

FOLLOW_14_in_signedAtom501

```
final BitSet parser.FormulaParser.FOLLOW_14_in_signedAtom501 [static]
```

Definition at line 1449 of file FormulaParser.java.

FOLLOW_15_in_multExpr139

```
final BitSet parser.FormulaParser.FOLLOW_15_in_multExpr139 [static]
```

Definition at line 1263 of file FormulaParser.java.

FOLLOW_16_in_function165

```
final BitSet parser.FormulaParser.FOLLOW_16_in_function165 [static]
```

Definition at line 1267 of file FormulaParser.java.

FOLLOW_17_in_function176

```
final BitSet parser.FormulaParser.FOLLOW_17_in_function176 [static]
```

Definition at line 1273 of file FormulaParser.java.

FOLLOW_18_in_function187

```
final BitSet parser.FormulaParser.FOLLOW_18_in_function187 [static]
```

Definition at line 1279 of file FormulaParser.java.

FOLLOW_19_in_function198

```
final BitSet parser.FormulaParser.FOLLOW_19_in_function198 [static]
```

Definition at line 1285 of file FormulaParser.java.

FOLLOW_20_in_function209

```
final BitSet parser.FormulaParser.FOLLOW_20_in_function209 [static]
```

Definition at line 1291 of file FormulaParser.java.

FOLLOW_21_in_function224

```
final BitSet parser.FormulaParser.FOLLOW_21_in_function224 [static]
```

Definition at line 1301 of file FormulaParser.java.

FOLLOW_22_in_function235

```
final BitSet parser.FormulaParser.FOLLOW_22_in_function235 [static]
```

Definition at line 1307 of file FormulaParser.java.

FOLLOW_23_in_function246

```
final BitSet parser.FormulaParser.FOLLOW_23_in_function246 [static]
```

Definition at line 1313 of file FormulaParser.java.

FOLLOW_24_in_function257

```
final BitSet parser.FormulaParser.FOLLOW_24_in_function257 [static]
```

Definition at line 1319 of file FormulaParser.java.

FOLLOW_25_in_function268

```
final BitSet parser.FormulaParser.FOLLOW_25_in_function268 [static]
```

Definition at line 1325 of file FormulaParser.java.

FOLLOW_26_in_function279

```
final BitSet parser.FormulaParser.FOLLOW_26_in_function279 [static]
```

Definition at line 1331 of file FormulaParser.java.

FOLLOW_27_in_function290

```
final BitSet parser.FormulaParser.FOLLOW_27_in_function290 [static]
```

Definition at line 1337 of file FormulaParser.java.

FOLLOW_28_in_function301

```
final BitSet parser.FormulaParser.FOLLOW_28_in_function301 [static]
```

Definition at line 1343 of file FormulaParser.java.

FOLLOW_29_in_function316

```
final BitSet parser.FormulaParser.FOLLOW_29_in_function316 [static]
```

Definition at line 1353 of file FormulaParser.java.

FOLLOW_30_in_function327

```
final BitSet parser.FormulaParser.FOLLOW_30_in_function327 [static]
```

Definition at line 1359 of file FormulaParser.java.

FOLLOW_31_in_function338

```
final BitSet parser.FormulaParser.FOLLOW_31_in_function338 [static]
```

Definition at line 1365 of file FormulaParser.java.

FOLLOW_32_in_function349

```
final BitSet parser.FormulaParser.FOLLOW_32_in_function349 [static]
```

Definition at line 1371 of file FormulaParser.java.

FOLLOW_33_in_function364

```
final BitSet parser.FormulaParser.FOLLOW_33_in_function364 [static]
```

Definition at line 1381 of file FormulaParser.java.

FOLLOW_34_in_function379

```
final BitSet parser.FormulaParser.FOLLOW_34_in_function379 [static]
```

Definition at line 1391 of file FormulaParser.java.

FOLLOW_35_in_function394

```
final BitSet parser.FormulaParser.FOLLOW_35_in_function394 [static]
```

Definition at line 1401 of file FormulaParser.java.

FOLLOW_36_in_function401

```
final BitSet parser.FormulaParser.FOLLOW_36_in_function401 [static]
```

Definition at line 1403 of file FormulaParser.java.

FOLLOW_37_in_function412

```
final BitSet parser.FormulaParser.FOLLOW_37_in_function412 [static]
```

Definition at line 1409 of file FormulaParser.java.

FOLLOW_38_in_function423

```
final BitSet parser.FormulaParser.FOLLOW_38_in_function423 [static]
```

Definition at line 1415 of file FormulaParser.java.

FOLLOW_39_in_function434

```
final BitSet parser.FormulaParser.FOLLOW_39_in_function434 [static]
```

Definition at line 1421 of file FormulaParser.java.

FOLLOW_40_in_function445

```
final BitSet parser.FormulaParser.FOLLOW_40_in_function445 [static]
```

Definition at line 1427 of file FormulaParser.java.

FOLLOW_41_in_function456

```
final BitSet parser.FormulaParser.FOLLOW_41_in_function456 [static]
```

Definition at line 1433 of file FormulaParser.java.

FOLLOW_42_in_function467

```
final BitSet parser.FormulaParser.FOLLOW_42_in_function467 [static]
```

Definition at line 1439 of file FormulaParser.java.

FOLLOW_8_in_atom537

```
final BitSet parser.FormulaParser.FOLLOW_8_in_atom537 [static]
```

Definition at line 1457 of file FormulaParser.java.

FOLLOW_9_in_atom541

```
final BitSet parser.FormulaParser.FOLLOW_9_in_atom541 [static]
```

Definition at line 1461 of file FormulaParser.java.

FOLLOW_9_in_function169

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function169 [static]
```

Definition at line 1271 of file FormulaParser.java.

FOLLOW_9_in_function180

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function180 [static]
```

Definition at line 1277 of file FormulaParser.java.

FOLLOW_9_in_function191

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function191 [static]
```

Definition at line 1283 of file FormulaParser.java.

FOLLOW_9_in_function202

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function202 [static]
```

Definition at line 1289 of file FormulaParser.java.

FOLLOW_9_in_function217

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function217 [static]
```

Definition at line 1299 of file FormulaParser.java.

FOLLOW_9_in_function228

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function228 [static]
```

Definition at line 1305 of file FormulaParser.java.

FOLLOW_9_in_function239

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function239 [static]
```

Definition at line 1311 of file FormulaParser.java.

FOLLOW_9_in_function250

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function250 [static]
```

Definition at line 1317 of file FormulaParser.java.

FOLLOW_9_in_function261

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function261 [static]
```

Definition at line 1323 of file FormulaParser.java.

FOLLOW_9_in_function272

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function272 [static]
```

Definition at line 1329 of file FormulaParser.java.

FOLLOW_9_in_function283

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function283 [static]
```

Definition at line 1335 of file FormulaParser.java.

FOLLOW_9_in_function294

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function294 [static]
```

Definition at line 1341 of file FormulaParser.java.

FOLLOW_9_in_function309

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function309 [static]
```

Definition at line 1351 of file FormulaParser.java.

FOLLOW_9_in_function320

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function320 [static]
```

Definition at line 1357 of file FormulaParser.java.

FOLLOW_9_in_function331

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function331 [static]
```

Definition at line 1363 of file FormulaParser.java.

FOLLOW_9_in_function342

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function342 [static]
```

Definition at line 1369 of file FormulaParser.java.

FOLLOW_9_in_function357

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function357 [static]
```

Definition at line 1379 of file FormulaParser.java.

FOLLOW_9_in_function372

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function372 [static]
```

Definition at line 1389 of file FormulaParser.java.

FOLLOW_9_in_function387

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function387 [static]
```

Definition at line 1399 of file FormulaParser.java.

FOLLOW_9_in_function405

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function405 [static]
```

Definition at line 1407 of file FormulaParser.java.

FOLLOW_9_in_function416

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function416 [static]
```

Definition at line 1413 of file FormulaParser.java.

FOLLOW_9_in_function427

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function427 [static]
```

Definition at line 1419 of file FormulaParser.java.

FOLLOW_9_in_function438

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function438 [static]
```

Definition at line 1425 of file FormulaParser.java.

FOLLOW_9_in_function449

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function449 [static]
```

Definition at line 1431 of file FormulaParser.java.

FOLLOW_9_in_function460

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function460 [static]
```

Definition at line 1437 of file FormulaParser.java.

FOLLOW_9_in_function471

```
final BitSet parser.FormulaParser.FOLLOW_9_in_function471 [static]
```

Definition at line 1443 of file FormulaParser.java.

FOLLOW_atom_in_signedAtom494

```
final BitSet parser.FormulaParser.FOLLOW_atom_in_signedAtom494 [static]
```

Definition at line 1447 of file FormulaParser.java.

FOLLOW_atom_in_signedAtom505

```
final BitSet parser.FormulaParser.FOLLOW_atom_in_signedAtom505 [static]
```

Definition at line 1451 of file FormulaParser.java.

FOLLOW_expr_in_atom539

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_atom539 [static]
```

Definition at line 1459 of file FormulaParser.java.

FOLLOW_expr_in_formula47

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_formula47 [static]
```

Definition at line 1241 of file FormulaParser.java.

FOLLOW_expr_in_function168

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function168 [static]
```

Definition at line 1269 of file FormulaParser.java.

FOLLOW_expr_in_function179

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function179 [static]
```

Definition at line 1275 of file FormulaParser.java.

FOLLOW_expr_in_function190

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function190 [static]
```

Definition at line 1281 of file FormulaParser.java.

FOLLOW_expr_in_function201

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function201 [static]
```

Definition at line 1287 of file FormulaParser.java.

FOLLOW_expr_in_function212

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function212 [static]
```

Definition at line 1293 of file FormulaParser.java.

FOLLOW_expr_in_function216

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function216 [static]
```

Definition at line 1297 of file FormulaParser.java.

FOLLOW_expr_in_function227

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function227 [static]
```

Definition at line 1303 of file FormulaParser.java.

FOLLOW_expr_in_function238

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function238 [static]
```

Definition at line 1309 of file FormulaParser.java.

FOLLOW_expr_in_function249

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function249 [static]
```

Definition at line 1315 of file FormulaParser.java.

FOLLOW_expr_in_function260

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function260 [static]
```

Definition at line 1321 of file FormulaParser.java.

FOLLOW_expr_in_function271

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function271 [static]
```

Definition at line 1327 of file FormulaParser.java.

FOLLOW_expr_in_function282

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function282 [static]
```

Definition at line 1333 of file FormulaParser.java.

FOLLOW_expr_in_function293

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function293 [static]
```

Definition at line 1339 of file FormulaParser.java.

FOLLOW_expr_in_function304

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function304 [static]
```

Definition at line 1345 of file FormulaParser.java.

FOLLOW_expr_in_function308

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function308 [static]
```

Definition at line 1349 of file FormulaParser.java.

FOLLOW_expr_in_function319

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function319 [static]
```

Definition at line 1355 of file FormulaParser.java.

FOLLOW_expr_in_function330

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function330 [static]
```

Definition at line 1361 of file FormulaParser.java.

FOLLOW_expr_in_function341

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function341 [static]
```

Definition at line 1367 of file FormulaParser.java.

FOLLOW_expr_in_function352

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function352 [static]
```

Definition at line 1373 of file FormulaParser.java.

FOLLOW_expr_in_function356

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function356 [static]
```

Definition at line 1377 of file FormulaParser.java.

FOLLOW_expr_in_function367

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function367 [static]
```

Definition at line 1383 of file FormulaParser.java.

FOLLOW_expr_in_function371

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function371 [static]
```

Definition at line 1387 of file FormulaParser.java.

FOLLOW_expr_in_function382

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function382 [static]
```

Definition at line 1393 of file FormulaParser.java.

FOLLOW_expr_in_function386

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function386 [static]
```

Definition at line 1397 of file FormulaParser.java.

FOLLOW_expr_in_function404

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function404 [static]
```

Definition at line 1405 of file FormulaParser.java.

FOLLOW_expr_in_function415

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function415 [static]
```

Definition at line 1411 of file FormulaParser.java.

FOLLOW_expr_in_function426

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function426 [static]
```

Definition at line 1417 of file FormulaParser.java.

FOLLOW_expr_in_function437

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function437 [static]
```

Definition at line 1423 of file FormulaParser.java.

FOLLOW_expr_in_function448

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function448 [static]
```

Definition at line 1429 of file FormulaParser.java.

FOLLOW_expr_in_function459

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function459 [static]
```

Definition at line 1435 of file FormulaParser.java.

FOLLOW_expr_in_function470

```
final BitSet parser.FormulaParser.FOLLOW_expr_in_function470 [static]
```

Definition at line 1441 of file FormulaParser.java.

FOLLOW_FLOAT_in_atom522

```
final BitSet parser.FormulaParser.FOLLOW_FLOAT_in_atom522 [static]
```

Definition at line 1453 of file FormulaParser.java.

FOLLOW_function_in_atom548

```
final BitSet parser.FormulaParser.FOLLOW_function_in_atom548 [static]
```

Definition at line 1463 of file FormulaParser.java.

FOLLOW_ID_in_atom529

```
final BitSet parser.FormulaParser.FOLLOW_ID_in_atom529 [static]
```

Definition at line 1455 of file FormulaParser.java.

FOLLOW_multExpr_in_expr73

```
final BitSet parser.FormulaParser.FOLLOW_multExpr_in_expr73 [static]
```

Definition at line 1247 of file FormulaParser.java.

FOLLOW_multExpr_in_expr84

```
final BitSet parser.FormulaParser.FOLLOW_multExpr_in_expr84 [static]
```

Definition at line 1251 of file FormulaParser.java.

FOLLOW_multExpr_in_expr95

```
final BitSet parser.FormulaParser.FOLLOW_multExpr_in_expr95 [static]
```

Definition at line 1255 of file FormulaParser.java.

FOLLOW_NEWLINE_in_formula49

```
final BitSet parser.FormulaParser.FOLLOW_NEWLINE_in_formula49 [static]
```

Definition at line 1243 of file FormulaParser.java.

FOLLOW_NEWLINE_in_formula56

```
final BitSet parser.FormulaParser.FOLLOW_NEWLINE_in_formula56 [static]
```

Definition at line 1245 of file FormulaParser.java.

FOLLOW_signedAtom_in_multExpr120

```
final BitSet parser.FormulaParser.FOLLOW_signedAtom_in_multExpr120 [static]
```

Definition at line 1257 of file FormulaParser.java.

FOLLOW_signedAtom_in_multExpr132

```
final BitSet parser.FormulaParser.FOLLOW_signedAtom_in_multExpr132 [static]
```

Definition at line 1261 of file FormulaParser.java.

FOLLOW_signedAtom_in_multExpr143

```
final BitSet parser.FormulaParser.FOLLOW_signedAtom_in_multExpr143 [static]
```

Definition at line 1265 of file FormulaParser.java.

ID

```
final int parser.FormulaParser.ID [static]
```

Definition at line 121 of file FormulaParser.java.

NEWLINE

```
final int parser.FormulaParser.NEWLINE [static]
```

Definition at line 122 of file FormulaParser.java.

T__10

```
final int parser.FormulaParser.T__10 [static]
```

Definition at line 86 of file FormulaParser.java.

T__11

```
final int parser.FormulaParser.T__11 [static]
```

Definition at line 87 of file FormulaParser.java.

T__12

```
final int parser.FormulaParser.T__12 [static]
```

Definition at line 88 of file FormulaParser.java.

T__13

```
final int parser.FormulaParser.T__13 [static]
```

Definition at line 89 of file FormulaParser.java.

T__14

```
final int parser.FormulaParser.T__14 [static]
```

Definition at line 90 of file FormulaParser.java.

T__15

```
final int parser.FormulaParser.T__15 [static]
```

Definition at line 91 of file FormulaParser.java.

T__16

```
final int parser.FormulaParser.T__16 [static]
```

Definition at line 92 of file FormulaParser.java.

T__17

```
final int parser.FormulaParser.T__17 [static]
```

Definition at line 93 of file FormulaParser.java.

T__18

```
final int parser.FormulaParser.T__18 [static]
```

Definition at line 94 of file FormulaParser.java.

T__19

```
final int parser.FormulaParser.T__19 [static]
```

Definition at line 95 of file FormulaParser.java.

T__20

```
final int parser.FormulaParser.T__20 [static]
```

Definition at line 96 of file FormulaParser.java.

T__21

```
final int parser.FormulaParser.T__21 [static]
```

Definition at line 97 of file FormulaParser.java.

T__22

```
final int parser.FormulaParser.T__22 [static]
```

Definition at line 98 of file FormulaParser.java.

T__23

```
final int parser.FormulaParser.T__23 [static]
```

Definition at line 99 of file FormulaParser.java.

T__24

```
final int parser.FormulaParser.T__24 [static]
```

Definition at line 100 of file FormulaParser.java.

T__25

```
final int parser.FormulaParser.T__25 [static]
```

Definition at line 101 of file FormulaParser.java.

T__26

```
final int parser.FormulaParser.T__26 [static]
```

Definition at line 102 of file FormulaParser.java.

T__27

```
final int parser.FormulaParser.T__27 [static]
```

Definition at line 103 of file FormulaParser.java.

T__28

```
final int parser.FormulaParser.T__28 [static]
```

Definition at line 104 of file FormulaParser.java.

T__29

```
final int parser.FormulaParser.T__29 [static]
```

Definition at line 105 of file FormulaParser.java.

T__30

```
final int parser.FormulaParser.T__30 [static]
```

Definition at line 106 of file FormulaParser.java.

T__31

```
final int parser.FormulaParser.T__31 [static]
```

Definition at line 107 of file FormulaParser.java.

T__32

```
final int parser.FormulaParser.T__32 [static]
```

Definition at line 108 of file FormulaParser.java.

T__33

```
final int parser.FormulaParser.T__33 [static]
```

Definition at line 109 of file FormulaParser.java.

T__34

```
final int parser.FormulaParser.T__34 [static]
```

Definition at line 110 of file FormulaParser.java.

T__35

```
final int parser.FormulaParser.T__35 [static]
```

Definition at line 111 of file FormulaParser.java.

T__36

```
final int parser.FormulaParser.T__36 [static]
```

Definition at line 112 of file FormulaParser.java.

T__37

```
final int parser.FormulaParser.T__37 [static]
```

Definition at line 113 of file FormulaParser.java.

T__38

```
final int parser.FormulaParser.T__38 [static]
```

Definition at line 114 of file FormulaParser.java.

T__39

```
final int parser.FormulaParser.T__39 [static]
```

Definition at line 115 of file FormulaParser.java.

T__40

```
final int parser.FormulaParser.T__40 [static]
```

Definition at line 116 of file FormulaParser.java.

T__41

```
final int parser.FormulaParser.T__41 [static]
```

Definition at line 117 of file FormulaParser.java.

T__42

```
final int parser.FormulaParser.T__42 [static]
```

Definition at line 118 of file FormulaParser.java.

T__8

```
final int parser.FormulaParser.T__8 [static]
```

Definition at line 84 of file FormulaParser.java.

T__9

```
final int parser.FormulaParser.T__9 [static]
```

Definition at line 85 of file FormulaParser.java.

tokenNames

```
final String [] parser.FormulaParser.tokenNames [static]
```

Definition at line 74 of file FormulaParser.java.

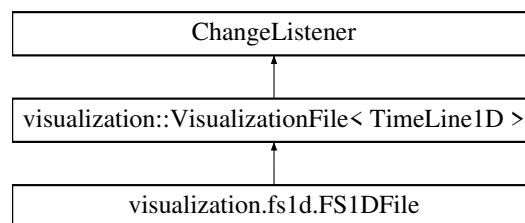
The documentation for this class was generated from the following file:

- src/parser/[FormulaParser.java](#)

6.46 visualization.fs1d.FS1DFile Class Reference

A FullSWOF_1D output file.

Inheritance diagram for visualization.fs1d.FS1DFile:

**Public Member Functions**

- [FS1DFile](#) (File file) throws IOException

Protected Member Functions

- void [setUpView](#) ()

Additional Inherited Members**6.46.1 Detailed Description**

A FullSWOF_1D output file.

Definition at line 69 of file FS1DFile.java.

6.46.2 Constructor & Destructor Documentation**FS1DFile()**

```
visualization.fs1d.FS1DFile.FS1DFile (
    File file ) throws IOException
```

Constructs a FullSWOF_1D file.

Parameters

<i>file</i>	the physical file
-------------	-------------------

Exceptions

<i>IOException</i>	if an error occurs that prevent the file from being read
--------------------	--

Definition at line 80 of file FS1DFile.java.

6.46.3 Member Function Documentation

setUpView()

```
void visualization.fs1d.FS1DFile.setUpView ( ) [protected]
```

Builds the file view

Definition at line 92 of file FS1DFile.java.

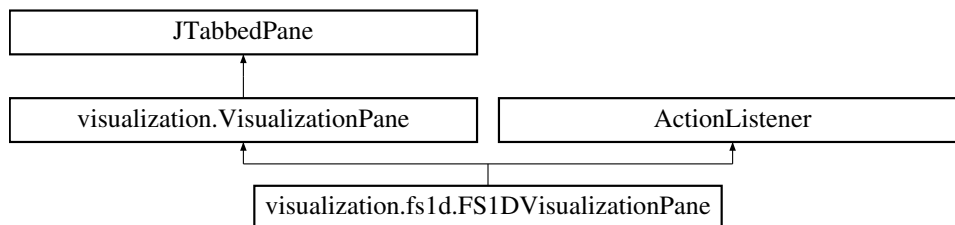
The documentation for this class was generated from the following file:

- src/visualization/fs1d/[FS1DFile.java](#)

6.47 visualization.fs1d.FS1DVisualizationPane Class Reference

A visualization pane for a FullSWOF_1D output file.

Inheritance diagram for visualization.fs1d.FS1DVisualizationPane:



Public Member Functions

- [FS1DVisualizationPane](#) ([FS1DFile](#) file)
- void [actionPerformed](#) (ActionEvent evt)
- void [update](#) ()

Additional Inherited Members

6.47.1 Detailed Description

A visualization pane for a FullSWOF_1D output file.

It includes an animated chart with spatial information and charts presenting the evolution of values at the boundaries.

Definition at line 87 of file FS1DVisualizationPane.java.

6.47.2 Constructor & Destructor Documentation

FS1DVisualizationPane()

```
visualization.fs1d.FS1DVisualizationPane.FS1DVisualizationPane (
    FS1DFile file )
```

Constructs a visualization pane.

Parameters

<i>file</i>	the file to represent
-------------	-----------------------

Definition at line 133 of file FS1DVisualizationPane.java.

6.47.3 Member Function Documentation**actionPerformed()**

```
void visualization.fs1d.FS1DVisualizationPane.actionPerformed (
    ActionEvent evt )
```

Called when the user interacts with one the combo boxes to change the type of information displayed on charts.

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 157 of file FS1DVisualizationPane.java.

update()

```
void visualization.fs1d.FS1DVisualizationPane.update ( )
```

Updates the charts to reflect changes on the timeline.

Definition at line 186 of file FS1DVisualizationPane.java.

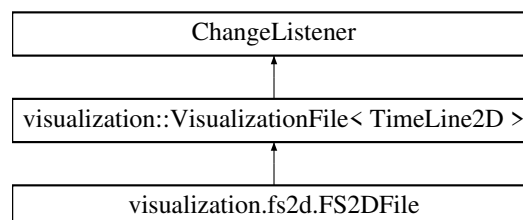
The documentation for this class was generated from the following file:

- src/visualization/fs1d/[FS1DVisualizationPane.java](#)

6.48 visualization.fs2d.FS2DFile Class Reference

A FullSWOF_2D output file.

Inheritance diagram for visualization.fs2d.FS2DFile:

**Classes**

- enum [Format](#)

The different formats of files produced by FullSWOF_2D.

Public Member Functions

- [FS2DFile](#) (File file, [Format](#) format) throws IOException

Protected Member Functions

- void [setUpView](#) ()

Additional Inherited Members

6.48.1 Detailed Description

A FullSWOF_2D output file.

Definition at line 69 of file FS2DFile.java.

6.48.2 Constructor & Destructor Documentation

FS2DFile()

```
visualization.fs2d.FS2DFile.FS2DFile (
    File file,
    Format format ) throws IOException
Constructs a FullSWOF_2D file.
```

Parameters

<i>file</i>	the physical file
<i>format</i>	the format of the physical file (Gnuplot or VTK)

Exceptions

<i>IOException</i>	if an error occurs that prevent the file from being read
--------------------	--

Definition at line 82 of file FS2DFile.java.

6.48.3 Member Function Documentation

setUpView()

```
void visualization.fs2d.FS2DFile.setUpView ( ) [protected]
```

Builds the file view

Definition at line 99 of file FS2DFile.java.

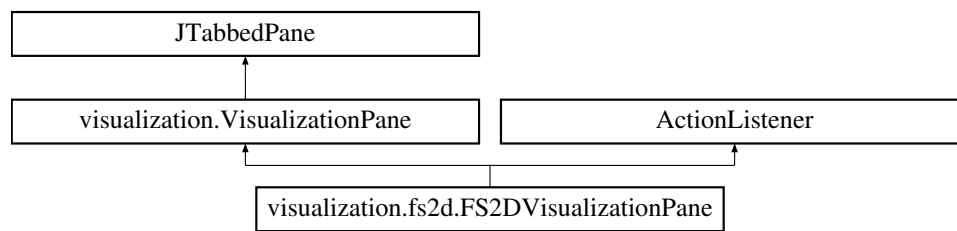
The documentation for this class was generated from the following file:

- src/visualization/fs2d/[FS2DFile.java](#)

6.49 visualization.fs2d.FS2DVisualizationPane Class Reference

A tabbed pane presenting a FullSWOF_2D File.

Inheritance diagram for visualization.fs2d.FS2DVisualizationPane:



Public Member Functions

- [FS2DVisualizationPane](#) ([FS2DFile](#) file)
- void [update](#) ()
- void [actionPerformed](#) (ActionEvent evt)

Additional Inherited Members

6.49.1 Detailed Description

A tabbed pane presenting a FullSWOF_2D File.

The first tab is an animated spatial representation. The other four tabs show the evolution during time at the boundaries ; these tabs are activated only if the file contains multiple time steps.

Definition at line 87 of file `FS2DVisualizationPane.java`.

6.49.2 Constructor & Destructor Documentation

FS2DVisualizationPane()

```
visualization.fs2d.FS2DVisualizationPane.FS2DVisualizationPane (
    FS2DFile file )
```

Constructs a visualization pane.

Parameters

<i>file</i>	the file to represent
-------------	-----------------------

Definition at line 153 of file `FS2DVisualizationPane.java`.

6.49.3 Member Function Documentation

actionPerformed()

```
void visualization.fs2d.FS2DVisualizationPane.actionPerformed (
   (ActionEvent) evt )
```

Called when the user interacts with one the combo boxes to change the type of information displayed on charts.

Parameters

<i>evt</i>	the triggering event
------------	----------------------

Definition at line 394 of file `FS2DVisualizationPane.java`.

update()

```
void visualization.fs2d.FS2DVisualizationPane.update ( )
```

Updates the charts to reflect changes on the timeline.
 Definition at line 177 of file FS2DVisualizationPane.java.
 The documentation for this class was generated from the following file:

- src/visualization/fs2d/[FS2DVisualizationPane.java](#)

6.50 io.FullswofIO Class Reference

This class provides static methods to handle the interactions with the C++ FullSWOF code.

Static Public Member Functions

- static void [exec](#) (String command, File workingDirectory, File outputDirectory)
- static void [setSimultaneousVisualization](#) (boolean simultaneous)
- static boolean [simultaneousVisualization](#) ()

6.50.1 Detailed Description

This class provides static methods to handle the interactions with the C++ FullSWOF code.
 Definition at line 76 of file FullswofIO.java.

6.50.2 Member Function Documentation**exec()**

```
static void io.FullswofIO.exec (
    String command,
    File workingDirectory,
    File outputDirectory ) [static]
```

Executes an external application. The application is launched in a separate thread so as to not block the main thread.

Parameters

<i>command</i>	the command used to launch the application
<i>workingDirectory</i>	the working directory of the application, or null if the subprocess should inherit the working directory of the current process
<i>outputDirectory</i>	the directory in which the output files are generated

Definition at line 109 of file FullswofIO.java.

setSimultaneousVisualization()

```
static void io.FullswofIO.setSimultaneousVisualization (
    boolean simultaneous ) [static]
```

Sets simultaneous visualization of files during their creation by FullSWOF

Parameters

<i>simultaneous</i>	if true, files are read and visualized while they are created by FullSWOF
---------------------	---

Definition at line 137 of file FullswofIO.java.

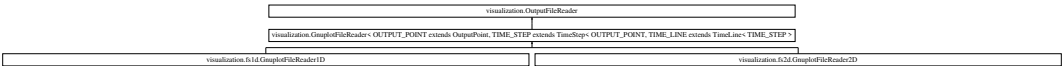
simultaneousVisualization()

static boolean io.FullswofIO.simultaneousVisualization () [static]
True if files visualized while they are created by FullSWOF
Definition at line 145 of file FullswofIO.java.
The documentation for this class was generated from the following file:

- src/io/[FullswofIO.java](#)

6.51 visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP > Class Template Reference

A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)
Inheritance diagram for visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >:



Public Member Functions

- [GnuplotFileReader](#) (File file, TIME_LINE [timeline](#)) throws IOException
- void [startWatching](#) ()
- void [stopWatching](#) ()
- void [updateTimeline](#) () throws IOException

Protected Member Functions

- abstract OUTPUT_POINT [makePoint](#) (String line)
- abstract TIME_STEP [makeTimeStep](#) ()
- TIME_STEP [readTimeStep](#) () throws IOException
- TIME_STEP [readTimeStep](#) (float t) throws IOException

Protected Attributes

- TIME_LINE [timeline](#)
The timeline being updated.

Additional Inherited Members

6.51.1 Detailed Description

A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)

Parameters

<OUTPUT_POINT>	the type of output points created by this reader
<TIME_STEP>	the type of time steps created by this reader
<TIME_LINE>	the type of timelines created by this reader

Definition at line 80 of file GnuplotFileReader.java.

6.51.2 Constructor & Destructor Documentation**GnuplotFileReader()**

```
visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends
TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.GnuplotFileReader (
    File file,
    TIME_LINE timeline ) throws IOException
```

Builds a Gnuplot file reader.

Parameters

<i>file</i>	the file to read
<i>timeline</i>	the timeline to update

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 114 of file GnuplotFileReader.java.

6.51.3 Member Function Documentation**makePoint()**

```
abstract OUTPUT_POINT visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint,
TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.make←
Point (
    String line ) [abstract], [protected]
```

Builds an output point from a line of file that describes it. The format of FullSWOF_1D and FullSWOF_2D are different so this method is implementation specific.

Parameters

<i>line</i>	a line of the file describing a single point
-------------	--

Returns

a cell from a FullSWOF output file.

makeTimeStep()

```
abstract TIME_STEP visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.makeTimeStep ( ) [abstract], [protected]
```

Builds an empty time step. Implementation specific.

Returns

an empty time step.

readTimeStep() [1/2]

```
TIME_STEP visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.readTimeStep ( ) throws IOException [protected]
```

Reads and builds a single time step.

Returns

a time step.

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 236 of file GnuplotFileReader.java.

readTimeStep() [2/2]

```
TIME_STEP visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.readTimeStep ( float t ) throws IOException [protected]
```

Reads and builds a single time step and give it a time value.

Parameters

<i>t</i>	the time value
----------	----------------

Returns

a time step.

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 268 of file GnuplotFileReader.java.

startWatching()

```
void visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends
TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.startWatching ( )
```

Start watching the physical for changes and update the timeline accordingly.

Definition at line 127 of file GnuplotFileReader.java.

stopWatching()

```
void visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends
TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.stopWatching ( )
```

Stop watching the physical file.

Definition at line 136 of file GnuplotFileReader.java.

updateTimeline()

```
void visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends
TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.updateTimeline ( ) throws
IOException
```

Modify the timeline so that its content reflects that of the physical file.

Exceptions

<i>IOException</i>	if an error occurs while reading
--------------------	----------------------------------

Definition at line 157 of file GnuplotFileReader.java.

6.51.4 Member Data Documentation**timeline**

```
TIME_LINE visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_S←
TEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >.timeline
[protected]
```

The timeline being updated.

Definition at line 101 of file GnuplotFileReader.java.

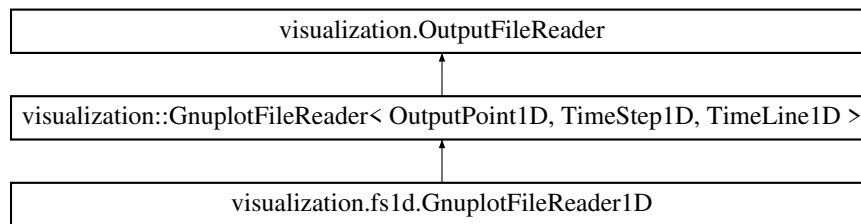
The documentation for this class was generated from the following file:

- src/visualization/[GnuplotFileReader.java](#)

6.52 visualization.fs1d.GnuplotFileReader1D Class Reference

A reader for Gnuplot output files generated by FullSWOF_1D.

Inheritance diagram for visualization.fs1d.GnuplotFileReader1D:



Public Member Functions

- [GnuplotFileReader1D](#) (File *file*, [TimeLine1D](#) *timeline*) throws IOException

Protected Member Functions

- [OutputPoint1D](#) *makePoint* (String *line*)
- [TimeStep1D](#) *makeTimeStep* ()

Additional Inherited Members

6.52.1 Detailed Description

A reader for Gnuplot output files generated by FullSWOF_1D.

Definition at line 71 of file GnuplotFileReader1D.java.

6.52.2 Constructor & Destructor Documentation

GnuplotFileReader1D()

```
visualization.fs1d.GnuplotFileReader1D.GnuplotFileReader1D (
    File file,
    TimeLine1D timeline ) throws IOException
```

Builds a reader for Gnuplot output files generated by FullSWOF_1D.

Parameters

<i>file</i>	the file to read
<i>timeline</i>	the timeline to update

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 85 of file GnuplotFileReader1D.java.

6.52.3 Member Function Documentation

makePoint()

```
OutputPoint1D visualization.fs1d.GnuplotFileReader1D.makePoint (
    String line ) [protected]
```

Builds an output point from a line of file that describes it.

Parameters

<i>line</i>	a line of the file describing a single point
-------------	--

Returns

a cell from a FullSWOF_1D output file.

Definition at line 98 of file GnuplotFileReader1D.java.

makeTimeStep()

`TimeStep1D visualization.fs1d.GnuplotFileReader1D.makeTimeStep ()` [protected]
`TimeStep1D` factory.

Returns

an empty `TimeStep1D`.

Definition at line 120 of file GnuplotFileReader1D.java.

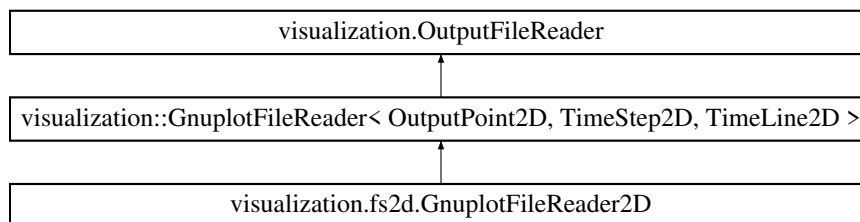
The documentation for this class was generated from the following file:

- src/visualization/fs1d/[GnuplotFileReader1D.java](#)

6.53 visualization.fs2d.GnuplotFileReader2D Class Reference

A reader for Gnuplot output files generated by FullSWOF_2D.

Inheritance diagram for visualization.fs2d.GnuplotFileReader2D:



Public Member Functions

- [GnuplotFileReader2D](#) (File file, [TimeLine2D](#) timeline) throws IOException

Protected Member Functions

- [OutputPoint2D](#) [makePoint](#) (String line)
- [TimeStep2D](#) [makeTimeStep](#) ()

Additional Inherited Members

6.53.1 Detailed Description

A reader for Gnuplot output files generated by FullSWOF_2D.

Definition at line 71 of file GnuplotFileReader2D.java.

6.53.2 Constructor & Destructor Documentation

GnuplotFileReader2D()

```
visualization.fs2d.GnuplotFileReader2D.GnuplotFileReader2D (
    File file,
    TimeLine2D timeline ) throws IOException
```

Builds a reader for Gnuplot output files generated by FullSWOF_2D.

Parameters

<i>file</i>	the file to read
<i>timeline</i>	the timeline to update

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 84 of file GnuplotFileReader2D.java.

6.53.3 Member Function Documentation

makePoint()

```
OutputPoint2D visualization.fs2d.GnuplotFileReader2D.makePoint (
    String line ) [protected]
```

Builds an output point from a line of file that describes it.

Parameters

<i>line</i>	a line of the file describing a single point
-------------	--

Returns

a cell from a FullSWOF_2D output file.

Definition at line 96 of file GnuplotFileReader2D.java.

makeTimeStep()

```
TimeStep2D visualization.fs2d.GnuplotFileReader2D.makeTimeStep ( ) [protected]
```

[TimeStep2D](#) factory.

Returns

an empty [TimeStep2D](#).

Definition at line 121 of file GnuplotFileReader2D.java.

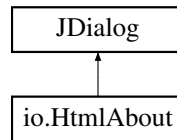
The documentation for this class was generated from the following file:

- src/visualization/fs2d/[GnuplotFileReader2D.java](#)

6.54 io.HtmlAbout Class Reference

An instance of this class is a JFrame used to display the content of About.

Inheritance diagram for io.HtmlAbout:



Public Member Functions

- [HtmlAbout](#) ()

6.54.1 Detailed Description

An instance of this class is a JFrame used to display the content of About.

It is useful to display the about

Definition at line 75 of file `HtmlAbout.java`.

6.54.2 Constructor & Destructor Documentation

HtmlAbout()

```
io.HtmlAbout.HtmlAbout ( )
```

Constructs a window displaying a about.

Definition at line 86 of file `HtmlAbout.java`.

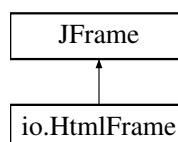
The documentation for this class was generated from the following file:

- `src/io/HtmlAbout.java`

6.55 io.HtmlFrame Class Reference

An instance of this class is a JFrame used to display the content of any HTML with basic style support.

Inheritance diagram for io.HtmlFrame:



Public Member Functions

- [HtmlFrame](#) (URL url, String title) throws IOException

6.55.1 Detailed Description

An instance of this class is a JFrame used to display the content of any HTML with basic style support.

It is useful to display the user manual or the application credits.

Definition at line 78 of file `HtmlFrame.java`.

6.55.2 Constructor & Destructor Documentation

HtmlFrame()

```
io.HtmlFrame.HtmlFrame (
    URL url,
    String title ) throws IOException
Constructs a window displaying a HTML file.
```

Parameters

<i>url</i>	the URL of the HTML file to display
<i>title</i>	the title of the frame

Exceptions

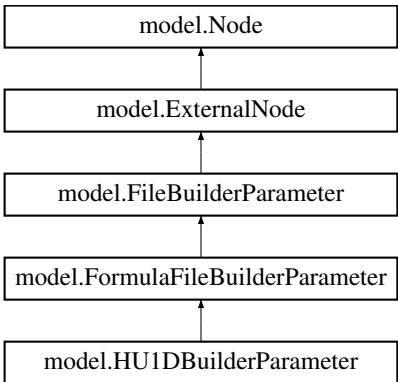
<i>IOException</i>	if the file cannot be read
--------------------	----------------------------

Definition at line 91 of file HtmlFrame.java.
The documentation for this class was generated from the following file:

- [src/io/HtmlFrame.java](#)

6.56 model.HU1DBuilderParameter Class Reference

A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u.
Inheritance diagram for model.HU1DBuilderParameter:



Public Member Functions

- [HU1DBuilderParameter](#) (String *name*, String *fileName*, [ExternalNode](#) *xLength*, [ExternalNode](#) *nxcells*)
- String [getFileContent](#) ()

Additional Inherited Members

6.56.1 Detailed Description

A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u.
Definition at line 70 of file HU1DBuilderParameter.java.

6.56.2 Constructor & Destructor Documentation

HU1DBuilderParameter()

```
model.HU1DBuilderParameter.HU1DBuilderParameter (
    String name,
    String fileName,
    ExternalNode xLength,
    ExternalNode nxcells )
```

Constructs a file builder that writes a HU file for FullSWOF_1D.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file generated
<i>xLength</i>	the node that indicates the value of the xLength parameter
<i>nxcells</i>	the node that indicates the value of the nxcells parameter

Definition at line 95 of file HU1DBuilderParameter.java.

6.56.3 Member Function Documentation

getFileContent()

```
String model.HU1DBuilderParameter.getFileContent ( )
```

Definition at line 106 of file HU1DBuilderParameter.java.

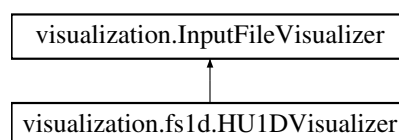
The documentation for this class was generated from the following file:

- src/model/[HU1DBuilderParameter.java](#)

6.57 visualization.fs1d.HU1DVisualizer Class Reference

A tool used to get a quick visualization of water input files for FullSWOF_1D.

Inheritance diagram for visualization.fs1d.HU1DVisualizer:



Public Member Functions

- [Chart getVisualization](#) (File file) throws IOException

6.57.1 Detailed Description

A tool used to get a quick visualization of water input files for FullSWOF_1D.

The file is represented as a chart with a line for water height and another for water velocity.

Definition at line 82 of file HU1DVisualizer.java.

6.57.2 Member Function Documentation

getVisualization()

`Chart visualization, fs1d.HU1DVisualizer.getVisualization (File file) throws IOException`

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).

Definition at line 84 of file HU1DVisualizer.java.

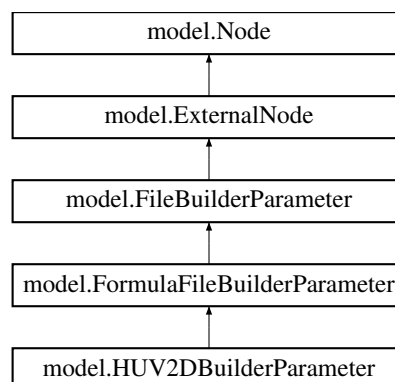
The documentation for this class was generated from the following file:

- `src/visualization/fs1d/HU1DVisualizer.java`

6.58 model.HUV2DBuilderParameter Class Reference

A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v.

Inheritance diagram for model.HUV2DBuilderParameter:



Public Member Functions

- `HUV2DBuilderParameter` (String `name`, String `fileName`, `ExternalNode` `xLength`, `ExternalNode` `nxcells`, `ExternalNode` `yLength`, `ExternalNode` `nycells`)
- String `getFileContent` ()

Additional Inherited Members

6.58.1 Detailed Description

A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v.

Definition at line 70 of file HUV2DBuilderParameter.java.

6.58.2 Constructor & Destructor Documentation

HUV2DBuilderParameter()

```
model.HUV2DBuilderParameter.HUV2DBuilderParameter (
    String name,
    String fileName,
    ExternalNode xLength,
    ExternalNode nxcells,
    ExternalNode yLength,
    ExternalNode nycells )
```

Constructs a file builder that writes a HUV file for FullSWOF_2D.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file generated
<i>xLength</i>	the node that indicates the value of the xLength parameter
<i>nxcells</i>	the node that indicates the value of the nxcells parameter
<i>yLength</i>	the node that indicates the value of the yLength parameter
<i>nycells</i>	the node that indicates the value of the nycells parameter

Definition at line 109 of file HUV2DBuilderParameter.java.

6.58.3 Member Function Documentation

getFileContent()

```
String model.HUV2DBuilderParameter.getFileContent ( )
```

Definition at line 121 of file HUV2DBuilderParameter.java.

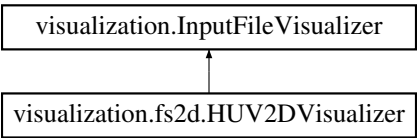
The documentation for this class was generated from the following file:

- src/model/[HUV2DBuilderParameter.java](#)

6.59 visualization.fs2d.HUV2DVisualizer Class Reference

A tool used to get a quick visualization of water input files for FullSWOF_2D.

Inheritance diagram for visualization.fs2d.HUV2DVisualizer:



Public Member Functions

- Component [getVisualization](#) (File file) throws IOException

6.59.1 Detailed Description

A tool used to get a quick visualization of water input files for FullSWOF_2D.
The file is represented as a 3D surface showing the water height. The color of the surface indicate the velocity of the water in each point.
Definition at line 86 of file HUV2DVisualizer.java.

6.59.2 Member Function Documentation

getVisualization()

Component `visualization.fs2d.HUV2DVisualizer.getVisualization (File file)` throws `IOException`
Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

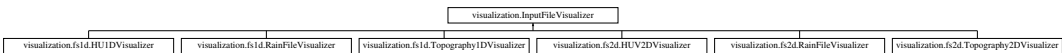
<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).
Definition at line 88 of file HUV2DVisualizer.java.
The documentation for this class was generated from the following file:

- `src/visualization/fs2d/HUV2DVisualizer.java`

6.60 visualization.InputFileVisualizer Interface Reference

A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)
Inheritance diagram for visualization.InputFileVisualizer:



Public Member Functions

- Component [getVisualization](#) (File file) throws IOException

6.60.1 Detailed Description

A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)

Definition at line 69 of file InputFileVisualizer.java.

6.60.2 Member Function Documentation

getVisualization()

Component `visualization.InputFileVisualizer.getVisualization (`
 File *file*) throws IOException

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implemented in [visualization.fs2d.Topography2DVisualizer](#), [visualization.fs2d.HUV2DVisualizer](#), [visualization.fs1d.HU1DVisualizer](#), [visualization.fs1d.RainFileVisualizer](#), [visualization.fs1d.Topography1DVisualizer](#), and [visualization.fs2d.RainFileVisualizer](#).

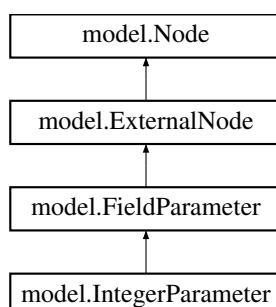
The documentation for this interface was generated from the following file:

- [src/visualization/InputFileVisualizer.java](#)

6.61 model.IntegerParameter Class Reference

A parameter with an integer value.

Inheritance diagram for model.IntegerParameter:



Public Member Functions

- `IntegerParameter` (String `name`, String `tag`)
- `IntegerParameter` (String `name`, String `tag`, String `description`)
- `IntegerParameter` (String `name`, String `tag`, `Interval` `valueInterval`)
- `IntegerParameter` (String `name`, String `tag`, String `description`, `Interval` `valueInterval`)
- boolean `isValid` ()
- `NodeController` `setUpController` ()

Additional Inherited Members

6.61.1 Detailed Description

A parameter with an integer value.

The value is still stored as a string, which is parsed to check if whether it represents a numeric value. The acceptance interval of the value can be specified in the constructor or omitted, in which case any integer will be considered valid.

Definition at line 73 of file `IntegerParameter.java`.

6.61.2 Constructor & Destructor Documentation

`IntegerParameter()` [1/4]

```
model.IntegerParameter.IntegerParameter (
    String name,
    String tag )
```

Constructs an integer parameter with the provided name and tag, and no description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node

Definition at line 91 of file `IntegerParameter.java`.

`IntegerParameter()` [2/4]

```
model.IntegerParameter.IntegerParameter (
    String name,
    String tag,
    String description )
```

Constructs an integer parameter with the provided name, tag and description.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node

Definition at line 107 of file `IntegerParameter.java`.

IntegerParameter() [3/4]

```
model.IntegerParameter.IntegerParameter (
    String name,
    String tag,
    Interval valueInterval )
```

Constructs an integer parameter with the provided name and tag, no description and an acceptance interval.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>valueInterval</i>	the interval in which the value is considered a valid entry

Definition at line 123 of file IntegerParameter.java.

IntegerParameter() [4/4]

```
model.IntegerParameter.IntegerParameter (
    String name,
    String tag,
    String description,
    Interval valueInterval )
```

Constructs a floating point number parameter with the provided name, tag, description and acceptance interval.

Parameters

<i>name</i>	the name of the node
<i>tag</i>	the tag of the node
<i>description</i>	a description of the node
<i>valueInterval</i>	the interval in which the value is considered a valid entry

Definition at line 142 of file IntegerParameter.java.

6.61.3 Member Function Documentation**isValid()**

```
boolean model.IntegerParameter.isValid ( )
```

Returns

true if the value string can be parsed to an integer within the parameter's acceptance interval.

See also

java.lang.Integer.valueOf(String s)

Definition at line 153 of file IntegerParameter.java.

setUpController()

`NodeController` `model.IntegerParameter.setUpController ()`

Builds a controller for this node.

Returns

the controller for this node.

See also

[ui.FieldParameterController](#)

Definition at line 178 of file `IntegerParameter.java`.

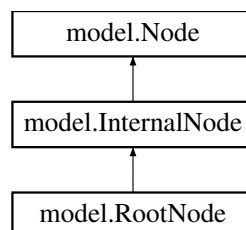
The documentation for this class was generated from the following file:

- `src/model/IntegerParameter.java`

6.62 model.InternalNode Class Reference

This class can be used for any internal node of the tree.

Inheritance diagram for `model.InternalNode`:

**Public Member Functions**

- `InternalNode` (String `name`)
- `InternalNode` (String `name`, String `description`)
- void `addChangeListener` (ChangeListener `c`)
- boolean `addNode` (Node `n`)
- boolean `fromFile` (File `file`) throws IOException
- List< Node > `getNodeList` ()
- boolean `isValid` ()
- `NodeController` `setUpController` ()
- String `toFile` (boolean `verbose`)
- void `updateChangeEvent` (String `tag`)

Package Attributes

- List< Node > `nodeList`

The list of child nodes.

Additional Inherited Members

6.62.1 Detailed Description

This class can be used for any internal node of the tree.

The standard controller instantiate the view as a panel.

See also

[ui.ParametersGroupController](#)

Definition at line 79 of file InternalNode.java.

6.62.2 Constructor & Destructor Documentation

InternalNode() [1/2]

```
model.InternalNode.InternalNode (
    String name )
```

Constructs an internal node with the provided name.

Parameters

<i>name</i>	the name of the node
-------------	----------------------

Definition at line 93 of file InternalNode.java.

InternalNode() [2/2]

```
model.InternalNode.InternalNode (
    String name,
    String description )
```

Constructs an internal node with the provided name and description.

Parameters

<i>name</i>	the name of the node
<i>description</i>	a description of the node

Definition at line 107 of file InternalNode.java.

6.62.3 Member Function Documentation

addChangeListener()

```
void model.InternalNode.addChangeListener (
    ChangeListener c )
```

Adds a change listener to the node. This is recursively forwarded to all child nodes, so that the change listener will be notified of changes on the node or any of its children.

Parameters

<i>c</i>	the change listener to be added
----------	---------------------------------

See also

`javax.swing.event.ChangeListener`

Definition at line 122 of file `InternalNode.java`.

addNode()

```
boolean model.InternalNode.addNode (
    Node n )
```

Adds a child node to the node.

Parameters

<i>n</i>	the child the be added
----------	------------------------

Returns

true if the child was successfully added.

Definition at line 135 of file `InternalNode.java`.

fromFile()

```
boolean model.InternalNode.fromFile (
    File file ) throws IOException
```

Attempts to set the values of the node from a file. This method is recursively forwarded to all child nodes, so that the values are set for the node and its children.

Parameters

<i>file</i>	the parameters file containing the values
-------------	---

Returns

true if all parameters were successfully found in the file and set in the model.

Exceptions

<i>IOException</i>	if a problem occurred while reading the file, such as the file not being found.
--------------------	---

Definition at line 153 of file `InternalNode.java`.

getNodeList()

```
List<Node> model.InternalNode.getNodeList ( )
```

Returns

a list of the child nodes.

Definition at line 166 of file InternalNode.java.

isValid()

```
boolean model.InternalNode.isValid ( )
```

Indicate whether the node is valid according to FullSWOF specifications. This method should is recursively forwarded to all child nodes, so it returns true only if all the child nodes are valid as well.

Returns

true if the node is valid.

Definition at line 178 of file InternalNode.java.

setUpController()

```
NodeController model.InternalNode.setUpController ( )
```

Returns

a ParametersGroupController instance for this node.

See also

[ui.ParametersGroupController](#)

Definition at line 191 of file InternalNode.java.

toFile()

```
String model.InternalNode.toFile (
    boolean verbose )
```

Returns a string to be written in a parameters.txt file. This method is recursively forwarded to all child nodes, so that the string contains the values of all the nodes.

Parameters

<i>verbose</i>	indicates whether the file should include descriptions of the nodes
----------------	---

Returns

a string to be written in a parameters.txt file.

Definition at line 206 of file InternalNode.java.

updateChangeEvent()

```
void model.InternalNode.updateChangeEvent (
    String tag )
```

Check, if the tag sought is present in node if true so there is an update events

Parameters

<i>tag</i>	the tag sought
------------	----------------

Definition at line 223 of file *InternalNode.java*.

6.62.4 Member Data Documentation**nodeList**

`List<Node> model.InternalNode.nodeList` [package]

The list of child nodes.

Definition at line 84 of file *InternalNode.java*.

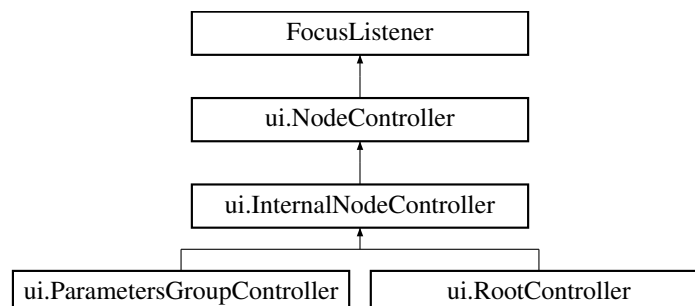
The documentation for this class was generated from the following file:

- `src/model/InternalNode.java`

6.63 *ui.InternalNodeController* Class Reference

The controller of an internal node.

Inheritance diagram for *ui.InternalNodeController*:

**Public Member Functions**

- [InternalNodeController](#) ([InternalNode](#) model)
- boolean [validate](#) (File mainDirectory)

Protected Member Functions

- boolean [addController](#) ([NodeController](#) c)

Package Attributes

- List< [NodeController](#) > [controllerList](#)

The list of the controllers of the controlled node children.

Additional Inherited Members

6.63.1 Detailed Description

The controller of an internal node.

The controllers maintain a hierarchy which is parallel to that of the model. The controller of an internal node must therefore maintain a list of child controllers, similar to the list of child nodes of its node.

Definition at line 77 of file `InternalNodeController.java`.

6.63.2 Constructor & Destructor Documentation

`InternalNodeController()`

```
ui.InternalNodeController.InternalNodeController (
    InternalNode model )
```

Constructs a controller for an internal node and all of its children, then instantiate a view for this node.

Parameters

<i>model</i>	the internal node to be controlled
--------------	------------------------------------

Definition at line 92 of file `InternalNodeController.java`.

6.63.3 Member Function Documentation

`addController()`

```
boolean ui.InternalNodeController.addController (
    NodeController c ) [protected]
```

Adds a controller to the controller children list. This method just be used by the constructor and only then.

Parameters

<i>c</i>	the controller to be added to the list
----------	--

Returns

true if the controller was successfully added.

Definition at line 129 of file `InternalNodeController.java`.

`validate()`

```
boolean ui.InternalNodeController.validate (
    File mainDirectory )
```

Applies validation procedures to the node. This method is called when a project using this node is saved or run. The method simply recursively forwards the call to the controller children.

Parameters

<i>mainDirectory</i>	a directory used by some validation procedure, it should usually be the project directory
----------------------	---

Returns

true if the node and all its children have been validated.

Definition at line 111 of file InternalNodeController.java.

6.63.4 Member Data Documentation

controllerList

```
List<NodeController> ui.InternalNodeController.controllerList [package]
```

The list of the controllers of the controlled node children.

Definition at line 82 of file InternalNodeController.java.

The documentation for this class was generated from the following file:

- [src/ui/InternalNodeController.java](#)

6.64 model.Interval Class Reference

Describes a numerical interval.

Public Member Functions

- [Interval](#) (float lowEndpoint, float highEndpoint)
- [Interval](#) (int lowEndpoint, int highEndpoint)
- [Interval](#) (float lowEndpoint, boolean lowEndpointIncluded, float highEndpoint, boolean highEndpointIncluded)
- [Interval](#) (int lowEndpoint, boolean lowEndpointIncluded, int highEndpoint, boolean highEndpointIncluded)
- boolean [isIncluded](#) (float f)
- boolean [isIncluded](#) (int i)

6.64.1 Detailed Description

Describes a numerical interval.

The interval is closed by default but the inclusion of each endpoint can be specified in the constructor

Definition at line 68 of file Interval.java.

6.64.2 Constructor & Destructor Documentation

Interval() [1/4]

```
model.Interval.Interval (
    float lowEndpoint,
    float highEndpoint )
```

Constructs a closed interval between the two endpoints (floats).

Parameters

<i>lowEndpoint</i>	the smallest of the two endpoints
<i>highEndpoint</i>	the largest of the two endpoints

Exceptions

<i>IllegalArgumentException</i>	if the low endpoint is greater than the high endpoint.
---------------------------------	--

Definition at line 103 of file Interval.java.

Interval() [2/4]

```
model.Interval.Interval (
    int lowEndpoint,
    int highEndpoint )
```

Constructs a closed interval between the two endpoints (ints).

Parameters

<i>lowEndpoint</i>	the smallest of the two endpoints
<i>highEndpoint</i>	the largest of the two endpoints

Exceptions

<i>IllegalArgumentException</i>	if the low endpoint is greater than the high endpoint.
---------------------------------	--

Definition at line 123 of file Interval.java.

Interval() [3/4]

```
model.Interval.Interval (
    float lowEndpoint,
    boolean lowEndpointIncluded,
    float highEndpoint,
    boolean highEndpointIncluded )
```

Constructs an interval between the two endpoints (floats).

Parameters

<i>lowEndpoint</i>	the smallest of the two endpoints
<i>lowEndpointIncluded</i>	indicates whether the low endpoint is part of the interval
<i>highEndpoint</i>	the largest of the two endpoints
<i>highEndpointIncluded</i>	indicates whether the high endpoint is part of the interval

Exceptions

<i>IllegalArgumentException</i>	if the low endpoint is greater than the high endpoint.
---------------------------------	--

Definition at line 147 of file Interval.java.

Interval() [4/4]

```
model.Interval.Interval (
```

```

    int lowEndpoint,
    boolean lowEndpointIncluded,
    int highEndpoint,
    boolean highEndpointIncluded )

```

Constructs an interval between the two endpoints (ints).

Parameters

<i>lowEndpoint</i>	the smallest of the two endpoints
<i>lowEndpointIncluded</i>	indicates whether the low endpoint is part of the interval
<i>highEndpoint</i>	the largest of the two endpoints
<i>highEndpointIncluded</i>	indicates whether the high endpoint is part of the interval

Exceptions

<i>IllegalArgumentException</i>	if the low endpoint is greater than the high endpoint.
---------------------------------	--

Definition at line 172 of file Interval.java.

6.64.3 Member Function Documentation

isIncluded() [1/2]

```

boolean model.Interval.isIncluded (
    float f )

```

Returns true if the float number is included in the interval

Parameters

<i>f</i>	the number to be tested
----------	-------------------------

Returns

true if *f* is included in the interval

Definition at line 190 of file Interval.java.

isIncluded() [2/2]

```

boolean model.Interval.isIncluded (
    int i )

```

Returns true if the integer is included in the interval

Parameters

<i>i</i>	the number to be tested
----------	-------------------------

Returns

true if *i* is included in the interval

Definition at line 210 of file Interval.java.

The documentation for this class was generated from the following file:

- src/model/[Interval.java](#)

6.65 visualization.JRealityViewingComponent Class Reference

This class provides a static method to build a viewing component for a JReality SceneGraphComponent.

Static Public Member Functions

- static Component [makeViewingComponent](#) (SceneGraphComponent sgc)

6.65.1 Detailed Description

This class provides a static method to build a viewing component for a JReality SceneGraphComponent.

The viewing component includes rotating, dragging and zooming tools as well as XYZ axes on the figure.

Definition at line 85 of file JRealityViewingComponent.java.

6.65.2 Member Function Documentation

makeViewingComponent()

```
static Component visualization.JRealityViewingComponent.makeViewingComponent (
    SceneGraphComponent sgc ) [static]
```

Builds a viewing component for a JReality sceneGraphComponent and returns it as a AWT component.

Parameters

<i>sgc</i>	the SceneGraphComponent to be viewed
------------	--------------------------------------

Returns

a AWT component.

Definition at line 96 of file JRealityViewingComponent.java.

The documentation for this class was generated from the following file:

- src/visualization/[JRealityViewingComponent.java](#)

6.66 model.ListFile Class Reference

The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : [RainFileParameter](#), PointFileParametre , BoundaryFileparameter.

Public Member Functions

- [ListFile](#) ()
- void [add](#) ([FileBuilderParameter](#) fileBuilderParameter, [MultipleChoiceParameter](#) multipleChoiceParameter, String tagValue)

- void `add` (`SettingDependency2` `settingDependency2`)
- void `check` ()
 Check if the table is valid.
- void `checkSaved` ()

6.66.1 Detailed Description

The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : `RainFileParameter`, `PointFileParametre` , `BoundaryFileparameter`.

Definition at line 70 of file `ListFile.java`.

6.66.2 Constructor & Destructor Documentation

ListFile()

```
model.ListFile.ListFile ( )
```

Definition at line 87 of file `ListFile.java`.

6.66.3 Member Function Documentation

add() [1/2]

```
void model.ListFile.add (
    FileBuilderParameter fileBuilderParameter,
    MultipleChoiceParameter multipleChoiceParameter,
    String tagValue )
```

Add tha table to check

Parameters

<i><code>fileBuilderParameter</code></i>	A parameter used to create an annex file
<i><code>multipleChoiceParameter</code></i>	the drop-down list associated at table
<i><code>tagValue</code></i>	parameter indicating that he is read from a file

Definition at line 104 of file `ListFile.java`.

add() [2/2]

```
void model.ListFile.add (
    SettingDependency2 settingDependency2 )
```

Definition at line 108 of file `ListFile.java`.

check()

```
void model.ListFile.check ( )
```

Check if the table is valid.

If the table isn't valid, so the drop-down list change the value So it activate the table instead of the file

Definition at line 117 of file `ListFile.java`.

checkSaved()

```
void model.ListFile.checkSaved ( )
```

Definition at line 125 of file ListFile.java.

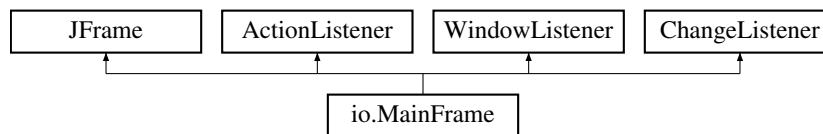
The documentation for this class was generated from the following file:

- [src/model/ListFile.java](#)

6.67 io.MainFrame Class Reference

An instance of this class is a JFrame corresponding to the main window of the user interface.

Inheritance diagram for io.MainFrame:

**Public Member Functions**

- [MainFrame](#) ()
- void [actionPerformed](#) (ActionEvent evt)
- void [stateChanged](#) (ChangeEvent e)
- void [updateContent](#) ()
- void [updateRecentFilesMenu](#) ()
- void [windowActivated](#) (WindowEvent arg0)
- void [windowClosed](#) (WindowEvent arg0)
- void [windowClosing](#) (WindowEvent arg0)
- void [windowDeactivated](#) (WindowEvent arg0)
- void [windowDeiconified](#) (WindowEvent arg0)
- void [windowIconified](#) (WindowEvent arg0)
- void [windowOpened](#) (WindowEvent arg0)

6.67.1 Detailed Description

An instance of this class is a JFrame corresponding to the main window of the user interface.

It includes the main menu and the area used to set the FullSWOF parameters.

See also

`javax.swing.JFrame`

Definition at line 88 of file MainFrame.java.

6.67.2 Constructor & Destructor Documentation**MainFrame()**

```
io.MainFrame.MainFrame ( )
```

Creates and displays a new main window

Definition at line 106 of file MainFrame.java.

6.67.3 Member Function Documentation

actionPerformed()

```
void io.MainFrame.actionPerformed (
    ActionEvent evt )
```

This method is called when a menu item is called. The resulting action depends on the command associated with the menu item.

Parameters

<i>evt</i>	the event associated with the menu item that fired a call to this method
------------	--

Definition at line 132 of file MainFrame.java.

stateChanged()

```
void io.MainFrame.stateChanged (
    ChangeEvent e )
```

Called when the state of the model (FullSWOF configuration) has changed

Definition at line 168 of file MainFrame.java.

updateContent()

```
void io.MainFrame.updateContent ( )
```

Update the content of the window. This method must be called when the model (FullSWOF configuration) has been changed. It is also called by the constructor.

Definition at line 178 of file MainFrame.java.

updateRecentFilesMenu()

```
void io.MainFrame.updateRecentFilesMenu ( )
```

Update the menu containing the recent files list. Each menu item is a file name with the action command "open "+File.getAbsolutePath().

Definition at line 189 of file MainFrame.java.

windowActivated()

```
void io.MainFrame.windowActivated (
    WindowEvent arg0 )
```

Called when the main window is activated. No effect.

Definition at line 207 of file MainFrame.java.

windowClosed()

```
void io.MainFrame.windowClosed (
    WindowEvent arg0 )
```

Called when the main window is closed. No effect.

Definition at line 214 of file MainFrame.java.

windowClosing()

```
void io.MainFrame.windowClosing (
    WindowEvent arg0 )
```

Called when the user attempts to close the window in any way. Displays a confirmation message before closing the window.

Definition at line 222 of file MainFrame.java.

windowDeactivated()

```
void io.MainFrame.windowDeactivated (
    WindowEvent arg0 )
```

Called when the main window is deactivated. No effect.

Definition at line 230 of file MainFrame.java.

windowDeiconified()

```
void io.MainFrame.windowDeiconified (
    WindowEvent arg0 )
```

Called when the main window is deiconified. No effect.

Definition at line 237 of file MainFrame.java.

windowIconified()

```
void io.MainFrame.windowIconified (
    WindowEvent arg0 )
```

Called when the main window is iconified. No effect.

Definition at line 244 of file MainFrame.java.

windowOpened()

```
void io.MainFrame.windowOpened (
    WindowEvent arg0 )
```

Called when the main window is opened. No effect.

Definition at line 251 of file MainFrame.java.

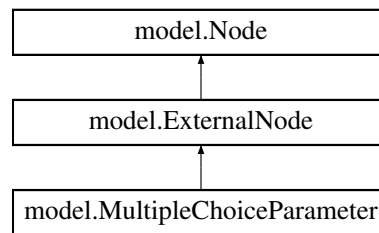
The documentation for this class was generated from the following file:

- [src/io/MainFrame.java](#)

6.68 model.MultipleChoiceParameter Class Reference

A parameter with a finite set of accepted values.

Inheritance diagram for model.MultipleChoiceParameter:



Classes

- class [PossibleValue](#)

A possible value is constituted of two strings.

Public Member Functions

- [MultipleChoiceParameter](#) (String [name](#), String [tag](#))
- [MultipleChoiceParameter](#) (String [name](#), String [tag](#), String [description](#))
- void [addPossibleValue](#) (String [name](#), String [value](#))
- void [addPossibleValue](#) (String [name](#), String [value](#), String [printedValue](#))
- String [toFile](#) (boolean [verbose](#))
- List< [PossibleValue](#) > [getPossibleValues](#) ()
- boolean [isValid](#) ()
- void [setValue](#) (String [newValue](#))
- [NodeController](#) [setUpController](#) ()

Additional Inherited Members

6.68.1 Detailed Description

A parameter with a finite set of accepted values.

Definition at line 70 of file `MultipleChoiceParameter.java`.

6.68.2 Constructor & Destructor Documentation

MultipleChoiceParameter() [1/2]

```

model.MultipleChoiceParameter.MultipleChoiceParameter (
    String name,
    String tag )
  
```

Constructs a multiple choice parameter with the provided name and tag, and no description. The parameter does not have any possible values upon construction, so you must add possible values to make it usable.

Parameters

<i>name</i>	the name of the parameter
<i>tag</i>	the tag of the parameter

Definition at line 88 of file `MultipleChoiceParameter.java`.

MultipleChoiceParameter() [2/2]

```
model.MultipleChoiceParameter.MultipleChoiceParameter (
    String name,
    String tag,
    String description )
```

Constructs a multiple choice parameter with the provided name, tag, and description. The parameter does not have any possible values upon construction, so you must add possible values to make it usable.

Parameters

<i>name</i>	the name of the parameter
<i>tag</i>	the tag of the parameter
<i>description</i>	a description of the parameter

Definition at line 106 of file MultipleChoiceParameter.java.

6.68.3 Member Function Documentation**addPossibleValue() [1/2]**

```
void model.MultipleChoiceParameter.addPossibleValue (
    String name,
    String value )
```

Adds a value to the set of possible values of the parameter.

Parameters

<i>name</i>	the display name of the value
<i>value</i>	the value as it will be written on the file

Definition at line 120 of file MultipleChoiceParameter.java.

addPossibleValue() [2/2]

```
void model.MultipleChoiceParameter.addPossibleValue (
    String name,
    String value,
    String printedValue )
```

Adds a value to the set of possible values of the parameter.

Parameters

<i>name</i>	the display name of the value
<i>value</i>	the value as it will be written on the file
<i>printedValue</i>	the value printed in the file, if it is different from value

Definition at line 135 of file MultipleChoiceParameter.java.

getPossibleValues()

```
List<PossibleValue> model.MultipleChoiceParameter.getPossibleValues ( )
```

Returns

the list of possible values for this parameter

Definition at line 169 of file MultipleChoiceParameter.java.

isValid()

```
boolean model.MultipleChoiceParameter.isValid ( )
```

Returns

true if the node dependencies are respected and the value is one the possible values

Definition at line 177 of file MultipleChoiceParameter.java.

setUpController()

```
NodeController model.MultipleChoiceParameter.setUpController ( )
```

Build a controller for this node.

Returns

the controller for this node.

See also

[ui.MultipleChoiceParameterController](#)

Definition at line 210 of file MultipleChoiceParameter.java.

setValue()

```
void model.MultipleChoiceParameter.setValue (
    String newValue )
```

Definition at line 193 of file MultipleChoiceParameter.java.

toFile()

```
String model.MultipleChoiceParameter.toFile (
    boolean verbose )
```

Returns a string to be written in a parameters.txt file. The string will be of the form <tag>:: value followed by an newline character. The value will be omitted if the node is disabled. The tag will be preceded by a description if verbose is set to true.

Parameters

<i>verbose</i>	indicates whether the file should include a description of the node
----------------	---

Returns

a string to be written in a parameters.txt file.

Definition at line 151 of file MultipleChoiceParameter.java.

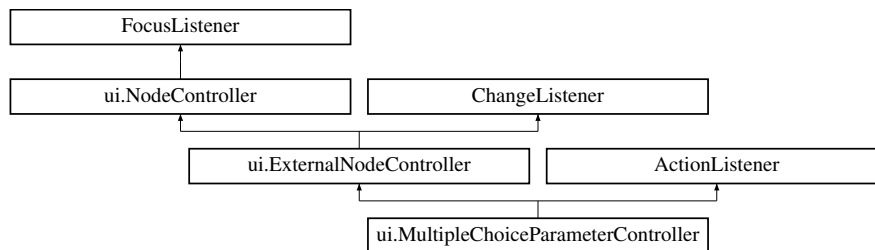
The documentation for this class was generated from the following file:

- [src/model/MultipleChoiceParameter.java](#)

6.69 ui.MultipleChoiceParameterController Class Reference

A controller for a multiple choice parameter node.

Inheritance diagram for ui.MultipleChoiceParameterController:



Public Member Functions

- [MultipleChoiceParameterController](#) ([MultipleChoiceParameter](#) model)
- void [actionPerformed](#) (ActionEvent arg0)
- void [highlightView](#) ()
- void [setUpView](#) ()
- void [updateModel](#) ()
- void [updateView](#) ()

Additional Inherited Members

6.69.1 Detailed Description

A controller for a multiple choice parameter node.

The view provided by this controller is made of a label and a combo box listing the possible values.

Definition at line 81 of file MultipleChoiceParameterController.java.

6.69.2 Constructor & Destructor Documentation

MultipleChoiceParameterController()

```
ui.MultipleChoiceParameterController.MultipleChoiceParameterController (
    MultipleChoiceParameter model )
```

Constructs a controller for a multiple choice parameter.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 101 of file MultipleChoiceParameterController.java.

6.69.3 Member Function Documentation

actionPerformed()

```
void ui.MultipleChoiceParameterController.actionPerformed (
    ActionEvent arg0 )
```

Called when an action is performed on the combo box. Updates the model.

Definition at line 116 of file MultipleChoiceParameterController.java.

highlightView()

```
void ui.MultipleChoiceParameterController.highlightView ( )
```

Brings the focus on the combo box.

Definition at line 125 of file MultipleChoiceParameterController.java.

setUpView()

```
void ui.MultipleChoiceParameterController.setUpView ( )
```

Sets up a view for this controller. The view is made of a label, followed by a combo box listing the possible values, on a single flowing line.

Definition at line 136 of file MultipleChoiceParameterController.java.

updateModel()

```
void ui.MultipleChoiceParameterController.updateModel ( )
```

Definition at line 156 of file MultipleChoiceParameterController.java.

updateView()

```
void ui.MultipleChoiceParameterController.updateView ( )
```

Definition at line 163 of file MultipleChoiceParameterController.java.

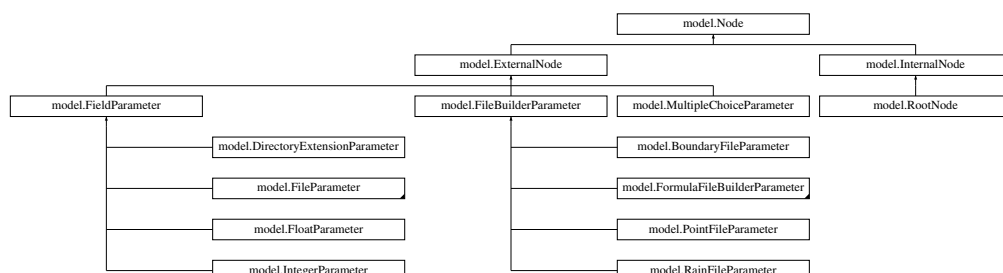
The documentation for this class was generated from the following file:

- [src/ui/MultipleChoiceParameterController.java](#)

6.70 model.Node Class Reference

A node in the model tree.

Inheritance diagram for model.Node:



Public Member Functions

- abstract void [addChangeListener](#) (ChangeListener c)
- abstract boolean [fromFile](#) (File file) throws IOException
- String [getName](#) ()
- abstract boolean [isValid](#) ()
- abstract [NodeController](#) [setUpController](#) ()
- abstract String [toFile](#) (boolean verbose)
- abstract void [updateChangeEvent](#) (String tag)
- String [toString](#) ()

Package Functions

- [Node](#) (String [name](#))
- [Node](#) (String [name](#), String [description](#))

Package Attributes

- String [name](#)
The name of the node.
- String [description](#)
A description of the node.

6.70.1 Detailed Description

A node in the model tree.

According to the composite design pattern of the model, this abstract class is extended by almost all classes from the package model, except for the [model.Dependency](#) class and classes that extend it.

Definition at line 77 of file Node.java.

6.70.2 Constructor & Destructor Documentation

Node() [1/2]

```
model.Node.Node (
    String name ) [package]
```

Constructs a node. The description for this node will be empty.

Parameters

<i>name</i>	the name of the node
-------------	----------------------

Definition at line 98 of file Node.java.

Node() [2/2]

```
model.Node.Node (
    String name,
    String description ) [package]
```

Constructs a node and the description provided.

Parameters

<i>name</i>	the name of the node
<i>description</i>	the description of this node

Definition at line 111 of file Node.java.

6.70.3 Member Function Documentation**addChangeListener()**

```
abstract void model.Node.addChangeListener (
    ChangeListener c ) [abstract]
```

Adds a change listener to the node.

Parameters

<i>c</i>	the change listener to be added
----------	---------------------------------

See also

javax.swing.event.ChangeListener

fromFile()

```
abstract boolean model.Node.fromFile (
    File file ) throws IOException [abstract]
```

Attempts to set the values of the node from a file.

Parameters

<i>file</i>	the parameters file containing the values
-------------	---

Returns

true if all parameters were successfully found in the file and set in the model.

Exceptions

<i>IOException</i>	if a problem occurred while reading the file, such as the file not being found.
--------------------	---

getName()

```
String model.Node.getName ( )
```

Returns

the name of the node.

Definition at line 143 of file Node.java.

isValid()

```
abstract boolean model.Node.isValid ( ) [abstract]
```

Indicate whether the node is valid according to FullSWOF specifications.

Returns

true if the node is valid

setUpController()

```
abstract NodeController model.Node.setUpController ( ) [abstract]
```

Builds a controller for the node. Each extending class must provide its own implementation of this method.

Returns

a node controller

toFile()

```
abstract String model.Node.toFile (
    boolean verbose ) [abstract]
```

Parameters

<i>verbose</i>	indicates whether the file should include descriptions of the nodes
----------------	---

Returns

a string to be written in a parameters.txt file.

toString()

```
String model.Node.toString ( )
```

Returns

the name of the node.

Definition at line 187 of file Node.java.

updateChangeEvent()

```
abstract void model.Node.updateChangeEvent (
    String tag ) [abstract]
```

Check, if the tag sought is equal to this instance if true so there is an update events

Parameters

<i>tag</i>	the tag sought
------------	----------------

6.70.4 Member Data Documentation**description**

```
String model.Node.description [package]
```

A description of the node.

This optional attribute is used mostly to provide a description of a parameter in the parameters.txt files generated by FullSWOF_UI

Definition at line 89 of file Node.java.

name

```
String model.Node.name [package]
```

The name of the node.

This name is used for display purposes in the views

Definition at line 82 of file Node.java.

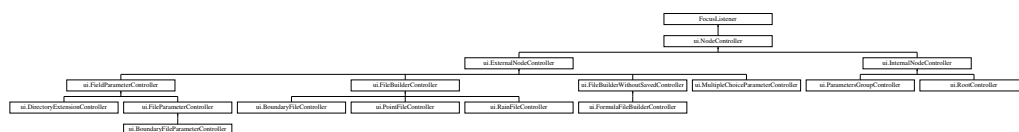
The documentation for this class was generated from the following file:

- [src/model/Node.java](#)

6.71 ui.NodeController Class Reference

A controller for a node, in the model-view-controller pattern.

Inheritance diagram for ui.NodeController:

**Public Member Functions**

- [NodeController](#) ([Node model](#))
- void [focusGained](#) (FocusEvent e)
- void [focusLost](#) (FocusEvent e)
- [Node getModel](#) ()
- JComponent [getView](#) ()
- abstract boolean [validate](#) (File mainDirectory)

Protected Member Functions

- void [resizeViewComponent](#) (JComponent c, int minWidth, int minHeight)

Static Protected Attributes

- static final Color [LIGHT_RED](#)
Color used for highlighting text fields and other view elements.
- static final int [MIN_LABEL_WIDTH](#)
A constant indicating the minimum width of field labels.
- static final int [MIN_FIELD_WIDTH](#)
A constant indicating the minimum width of text fields.
- static final int [MIN_BOX_WIDTH](#)
A constant indicating the minimum width of combo boxes.

Package Functions

- abstract void [setUpView](#) ()
Instantiate the view for this controller.

Package Attributes

- [Node model](#)
The node controlled by this controller.
- JComponent [view](#)
The view corresponding to this controller.

6.71.1 Detailed Description

A controller for a node, in the model-view-controller pattern.

The view is an attribute of the controller, and must be instantiated by the controller itself with a call to [setUpView\(\)](#).

See also

MVC pattern

Definition at line 80 of file NodeController.java.

6.71.2 Constructor & Destructor Documentation

NodeController()

```
ui.NodeController.NodeController (
    Node model )
```

Constructs a controller for a node.

Parameters

<i>model</i>	the controlled node
--------------	---------------------

Definition at line 119 of file NodeController.java.

6.71.3 Member Function Documentation

focusGained()

```
void ui.NodeController.focusGained (
    FocusEvent e )
```

Invoked when the view for this controller gains focus.

Definition at line 127 of file NodeController.java.

focusLost()

```
void ui.NodeController.focusLost (
    FocusEvent e )
```

Invoked when the view for this controller loses focus.

Definition at line 134 of file NodeController.java.

getModel()

```
Node ui.NodeController.getModel ( )
```

Returns

the node controlled by this controller.

Definition at line 141 of file NodeController.java.

getView()

```
JComponent ui.NodeController.getView ( )
```

Returns the view corresponding to this controller. It can be null if no [setUpView\(\)](#) has not been called.

Returns

the view corresponding to this controller

Definition at line 152 of file NodeController.java.

resizeViewComponent()

```
void ui.NodeController.resizeViewComponent (
    JComponent c,
    int minWidth,
    int minHeight ) [protected]
```

Resizes a JComponent to a minimum width and height. The component will be larger than those value if its preferred size is originally larger.

Parameters

<i>c</i>	the component to be resized
<i>minWidth</i>	the minimum width that the component should have
<i>minHeight</i>	the minimum height that the component should have

Definition at line 185 of file NodeController.java.

setUpView()

```
abstract void ui.NodeController.setUpView ( ) [abstract], [package]
```

Instantiate the view for this controller.

validate()

```
abstract boolean ui.NodeController.validate (
    File mainDirectory ) [abstract]
```

Applies validation procedures to the node. This method is called when a project using this node is saved or run.

Parameters

<i>mainDirectory</i>	a directory used by some validation procedure, it should usually be the project directory
----------------------	---

Returns

true if the node has been validated.

6.71.4 Member Data Documentation**LIGHT_RED**

```
final Color ui.NodeController.LIGHT_RED [static], [protected]
```

Color used for highlighting text fields and other view elements.
Definition at line 85 of file NodeController.java.

MIN_BOX_WIDTH

```
final int ui.NodeController.MIN_BOX_WIDTH [static], [protected]
```

A constant indicating the minimum width of combo boxes.
Definition at line 100 of file NodeController.java.

MIN_FIELD_WIDTH

```
final int ui.NodeController.MIN_FIELD_WIDTH [static], [protected]
```

A constant indicating the minimum width of text fields.
Definition at line 95 of file NodeController.java.

MIN_LABEL_WIDTH

```
final int ui.NodeController.MIN_LABEL_WIDTH [static], [protected]
```

A constant indicating the minimum width of field labels.
Definition at line 90 of file NodeController.java.

model

`Node ui.NodeController.model [package]`

The node controlled by this controller.

Definition at line 105 of file NodeController.java.

view

`JComponent ui.NodeController.view [package]`

The view corresponding to this controller.

Definition at line 110 of file NodeController.java.

The documentation for this class was generated from the following file:

- `src/ui/NodeController.java`

6.72 visualization.OutputFileReader Class Reference

An abstract class to implement readers for FullSWOF output files.

Inheritance diagram for visualization.OutputFileReader:

**Public Member Functions**

- void `addChangeListener` (ChangeListener c)
- void `removeChangeListener` (ChangeListener c)
- abstract void `startWatching` ()
- abstract void `stopWatching` ()
- abstract void `updateTimeline` () throws IOException

Protected Member Functions

- void `fireChangeEvent` ()

Protected Attributes

- `Timeline`<?> `timeline`

The timeline that the reader update.

Package Attributes

- ChangeEvent `changeEvent`
Used to lazily create change events.
- EventListenerList `listenerList`
The list of objects listening to event fired by this reader.

6.72.1 Detailed Description

An abstract class to implement readers for FullSWOF output files.

A reader should be able to update a timeline according to the data in the file. It can do this with a complete file or update the timeline during modifications of the physical file.

Definition at line 75 of file OutputFileReader.java.

6.72.2 Member Function Documentation

addChangeListener()

```
void visualization.OutputFileReader.addChangeListener (
    ChangeListener c )
```

Adds a change listener to the reader. Change listener should be notified of any change on the timeline.

Parameters

<i>c</i>	the change listener to be added
----------	---------------------------------

Definition at line 104 of file OutputFileReader.java.

fireChangeEvent()

```
void visualization.OutputFileReader.fireChangeEvent ( ) [protected]
```

Must be invoked when the timeline is modified.

See also

`javax.swing.event.EventListenerList`

Definition at line 148 of file OutputFileReader.java.

removeChangeListener()

```
void visualization.OutputFileReader.removeChangeListener (
    ChangeListener c )
```

Removes a change listener from the change listeners list of this reader.

Parameters

<i>c</i>	the change listener to be removed
----------	-----------------------------------

Definition at line 115 of file OutputFileReader.java.

startWatching()

```
abstract void visualization.OutputFileReader.startWatching ( ) [abstract]
```

Starts watching the physical for changes and update the timeline accordingly.

stopWatching()

```
abstract void visualization.OutputFileReader.stopWatching ( ) [abstract]
```

Stops watching the physical file.

updateTimeline()

```
abstract void visualization.OutputFileReader.updateTimeline ( ) throws IOException [abstract]
```

Modifies the timeline so that its content reflects that of the physical file.

Exceptions

<i>IOException</i>	if an error occurs while reading
--------------------	----------------------------------

6.72.3 Member Data Documentation**changeEvent**

`ChangeEvent visualization.OutputFileReader.changeEvent [package]`

Used to lazily create change events.

See also

`javax.swing.event.EventListenerList`

Definition at line 87 of file `OutputFileReader.java`.

listenerList

`EventListenerList visualization.OutputFileReader.listenerList [package]`

The list of objects listening to event fired by this reader.

See also

`javax.swing.event.EventListenerList`

Definition at line 94 of file `OutputFileReader.java`.

timeline

`Timeline<?> visualization.OutputFileReader.timeline [protected]`

The timeline that the reader update.

Definition at line 80 of file `OutputFileReader.java`.

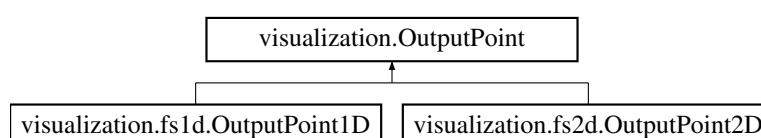
The documentation for this class was generated from the following file:

- `src/visualization/OutputFileReader.java`

6.73 visualization.OutputPoint Class Reference

A cell in a FullSWOF output file.

Inheritance diagram for `visualization.OutputPoint`:



6.73.1 Detailed Description

A cell in a FullSWOF output file.

Definition at line 64 of file OutputPoint.java.

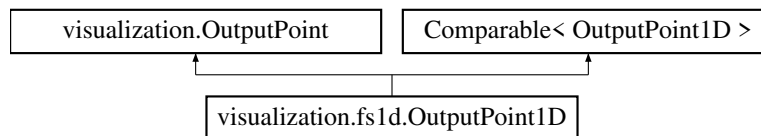
The documentation for this class was generated from the following file:

- [src/visualization/OutputPoint.java](#)

6.74 visualization.fs1d.OutputPoint1D Class Reference

A cell in a FullSWOF_1D output file.

Inheritance diagram for visualization.fs1d.OutputPoint1D:



Public Member Functions

- int [compareTo](#) ([OutputPoint1D](#) o)
- boolean [equals](#) (Object obj)
- float [getDx](#) ()
- float [getFr](#) ()
- float [getH](#) ()
- float [getHZ](#) ()
- float [getQ](#) ()
- float [getU](#) ()
- float [getZ](#) ()

Protected Member Functions

- [OutputPoint1D](#) (float dx, float h, float u, float z)

6.74.1 Detailed Description

A cell in a FullSWOF_1D output file.

Definition at line 65 of file OutputPoint1D.java.

6.74.2 Constructor & Destructor Documentation

OutputPoint1D()

```
visualization.fs1d.OutputPoint1D.OutputPoint1D (
    float dx,
    float h,
    float u,
    float z ) [protected]
```

Builds a cell.

Parameters

<i>dx</i>	the x coordinate of the point
<i>h</i>	the free surface (water height) at that point
<i>u</i>	the water velocity at that point
<i>z</i>	the topographic height at that point

Definition at line 100 of file OutputPoint1D.java.

6.74.3 Member Function Documentation**compareTo()**

```
int visualization.fs1d.OutputPoint1D.compareTo (
    OutputPoint1D o )
```

Cells are compared according to their x coordinate (ascending order).

Definition at line 112 of file OutputPoint1D.java.

equals()

```
boolean visualization.fs1d.OutputPoint1D.equals (
    Object obj )
```

Only the dx attribute is taken into account for equality.

Definition at line 127 of file OutputPoint1D.java.

getDx()

```
float visualization.fs1d.OutputPoint1D.getDx ( )
```

Returns

the x coordinate of the point.

Definition at line 143 of file OutputPoint1D.java.

getFr()

```
float visualization.fs1d.OutputPoint1D.getFr ( )
```

Returns

the Froude number of the point.

Definition at line 150 of file OutputPoint1D.java.

getH()

```
float visualization.fs1d.OutputPoint1D.getH ( )
```

Returns

the free surface (water height) at that point.

Definition at line 161 of file OutputPoint1D.java.

getHZ()

```
float visualization.fs1d.OutputPoint1D.getHZ ( )
```

Returns the total height at that point (water and topography).

Returns

the total height at that point.

Definition at line 170 of file OutputPoint1D.java.

getQ()

```
float visualization.fs1d.OutputPoint1D.getQ ( )
```

Returns

the discharge at that point.

Definition at line 177 of file OutputPoint1D.java.

getU()

```
float visualization.fs1d.OutputPoint1D.getU ( )
```

Returns

the water velocity at that point.

Definition at line 184 of file OutputPoint1D.java.

getZ()

```
float visualization.fs1d.OutputPoint1D.getZ ( )
```

Returns

the topographic height at that point.

Definition at line 191 of file OutputPoint1D.java.

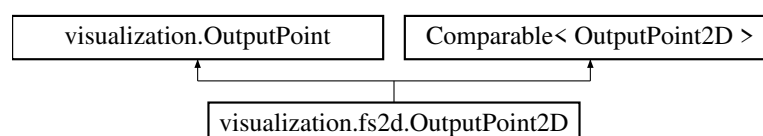
The documentation for this class was generated from the following file:

- src/visualization/fs1d/[OutputPoint1D.java](#)

6.75 visualization.fs2d.OutputPoint2D Class Reference

A cell in a FullSWOF_2D output file.

Inheritance diagram for visualization.fs2d.OutputPoint2D:



Public Member Functions

- int [compareTo](#) ([OutputPoint2D](#) o)
- boolean [equals](#) (Object obj)
- float [getDx](#) ()
- float [getDy](#) ()
- float [getFr](#) ()
- float [getH](#) ()
- float [getHZ](#) ()
- float [getN](#) ()
- float [getQ](#) ()
- float [getQx](#) ()
- float [getQy](#) ()
- float [getU](#) ()
- float [getV](#) ()
- float [getZ](#) ()

Protected Member Functions

- [OutputPoint2D](#) (float [dx](#), float [dy](#), float [h](#), float [u](#), float [v](#), float [z](#))

Package Attributes

- float [dx](#)
The x coordinate of the point.
- float [dy](#)
The y coordinate of the point.
- float [h](#)
The free surface (water height) at that point.
- float [u](#)
The water velocity in the x direction at that point.
- float [v](#)
The water velocity in the y direction at that point.
- float [z](#)
The topographic height at that point.

6.75.1 Detailed Description

A cell in a FullSWOF_2D output file.

Definition at line 66 of file OutputPoint2D.java.

6.75.2 Constructor & Destructor Documentation

OutputPoint2D()

```
visualization.fs2d.OutputPoint2D.OutputPoint2D (
    float dx,
    float dy,
    float h,
    float u,
    float v,
    float z ) [protected]
```

Builds a cell.

Parameters

<i>dx</i>	the x coordinate of the point
<i>dy</i>	the y coordinate of the point
<i>h</i>	the free surface (water height) at that point
<i>u</i>	the water velocity in the x direction at that point
<i>v</i>	the water velocity in the x direction at that point
<i>z</i>	the topographic height at that point

Definition at line 120 of file OutputPoint2D.java.

6.75.3 Member Function Documentation**compareTo()**

```
int visualization.fs2d.OutputPoint2D.compareTo (
    OutputPoint2D o )
```

Cells are compared according to their x coordinate (ascending order) then their y coordinate (ascending order).

Definition at line 136 of file OutputPoint2D.java.

equals()

```
boolean visualization.fs2d.OutputPoint2D.equals (
    Object obj )
```

Only the dx and dy attribute are taken into account for equality.

Definition at line 158 of file OutputPoint2D.java.

getDx()

```
float visualization.fs2d.OutputPoint2D.getDx ( )
```

Returns

the x coordinate of the point.

Definition at line 176 of file OutputPoint2D.java.

getDy()

```
float visualization.fs2d.OutputPoint2D.getDy ( )
```

Returns

the y coordinate of the point.

Definition at line 183 of file OutputPoint2D.java.

getFr()

```
float visualization.fs2d.OutputPoint2D.getFr ( )
```

Returns

the Froude number at the point.

Definition at line 190 of file OutputPoint2D.java.

getH()

```
float visualization.fs2d.OutputPoint2D.getH ( )
```

Returns

the free surface at the point.

Definition at line 201 of file OutputPoint2D.java.

getHZ()

```
float visualization.fs2d.OutputPoint2D.getHZ ( )
```

Returns the total height at the point (water + topography).

Returns

the total height at the point.

Definition at line 210 of file OutputPoint2D.java.

getN()

```
float visualization.fs2d.OutputPoint2D.getN ( )
```

Returns

the Euclidean norm of the water velocity at that point.

Definition at line 217 of file OutputPoint2D.java.

getQ()

```
float visualization.fs2d.OutputPoint2D.getQ ( )
```

Returns

the Euclidean norm of the discharge at that point.

Definition at line 224 of file OutputPoint2D.java.

getQx()

```
float visualization.fs2d.OutputPoint2D.getQx ( )
```

Returns

discharge in the x direction at that point.

Definition at line 231 of file OutputPoint2D.java.

getQy()

```
float visualization.fs2d.OutputPoint2D.getQy ( )
```

Returns

the discharge in the y direction at that point.

Definition at line 238 of file OutputPoint2D.java.

getU()

```
float visualization.fs2d.OutputPoint2D.getU ( )
```

Returns

the water velocity in the x direction at that point

Definition at line 245 of file OutputPoint2D.java.

getV()

```
float visualization.fs2d.OutputPoint2D.getV ( )
```

Returns

the water velocity in the x direction at that point.

Definition at line 252 of file OutputPoint2D.java.

getZ()

```
float visualization.fs2d.OutputPoint2D.getZ ( )
```

Returns

the topographic height at that point.

Definition at line 259 of file OutputPoint2D.java.

6.75.4 Member Data Documentation

dx

```
float visualization.fs2d.OutputPoint2D.dx [package]
```

The x coordinate of the point.

Definition at line 76 of file OutputPoint2D.java.

dy

```
float visualization.fs2d.OutputPoint2D.dy [package]
```

The y coordinate of the point.

Definition at line 81 of file OutputPoint2D.java.

h

```
float visualization.fs2d.OutputPoint2D.h [package]
```

The free surface (water height) at that point.

Definition at line 86 of file OutputPoint2D.java.

u

```
float visualization.fs2d.OutputPoint2D.u [package]
```

The water velocity in the x direction at that point.

Definition at line 91 of file OutputPoint2D.java.

v

```
float visualization.fs2d.OutputPoint2D.v [package]
```

The water velocity in the y direction at that point.

Definition at line 96 of file OutputPoint2D.java.

z

```
float visualization.fs2d.OutputPoint2D.z [package]
```

The topographic height at that point.

Definition at line 101 of file OutputPoint2D.java.

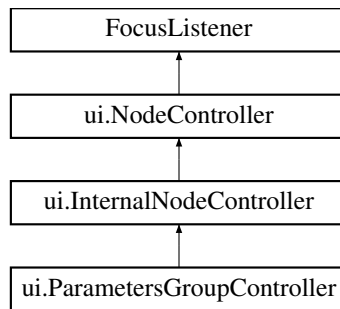
The documentation for this class was generated from the following file:

- src/visualization/fs2d/[OutputPoint2D.java](#)

6.76 ui.ParametersGroupController Class Reference

A controller for an internal node.

Inheritance diagram for ui.ParametersGroupController:



Public Member Functions

- [ParametersGroupController](#) ([InternalNode](#) node)

Package Functions

- void [setUpView](#) ()

Additional Inherited Members

6.76.1 Detailed Description

A controller for an internal node.

This controller can be used for any internal node, but [ui.RootController](#) might be better suited for the root of the tree. The difference between the two is only the view provided. This class provides a simple panel view with each child node on the same panel.

See also

[ui.RootController](#)

Definition at line 81 of file `ParametersGroupController.java`.

6.76.2 Constructor & Destructor Documentation

ParametersGroupController()

```
ui.ParametersGroupController.ParametersGroupController (
    InternalNode node )
```

Constructs a controller for an internal node.

Parameters

<i>node</i>	the internal node to be controlled.
-------------	-------------------------------------

Definition at line 88 of file `ParametersGroupController.java`.

6.76.3 Member Function Documentation

setUpView()

```
void ui.ParametersGroupController.setUpView ( ) [package]
```

Instantiates a panel view with each child node on the same panel.

Definition at line 97 of file ParametersGroupController.java.

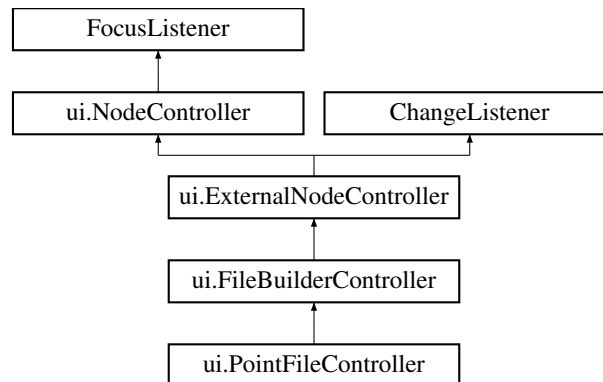
The documentation for this class was generated from the following file:

- [src/ui/ParametersGroupController.java](#)

6.77 ui.PointFileController Class Reference

A controller for a point file builder node.

Inheritance diagram for ui.PointFileController:



Classes

- class [CellRenderer](#)

An instance of this class is used to render the cells of the table in the view.

- class [PointModel](#)

The model used by the view table.

Public Member Functions

- [PointFileController](#) ([PointFileParameter](#) model)
- void [highlightView](#) ()
- void [updateModel](#) ()
- void [updateView](#) ()
- void [setUpView](#) ()

Package Attributes

- JTable [viewTable](#)

The table of the view.

- JLabel [viewLabel](#)

The label of the view.

Additional Inherited Members

6.77.1 Detailed Description

A controller for a point file builder node.

This controller can set up a view that includes an editable table where the user can write x and y value.

Definition at line 77 of file PointFileController.java.

6.77.2 Constructor & Destructor Documentation

PointFileController()

```
ui.PointFileController.PointFileController (
    PointFileParameter model )
```

Constructs a controller for a PointFileParameter.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 107 of file PointFileController.java.

6.77.3 Member Function Documentation

highlightView()

```
void ui.PointFileController.highlightView ( )
```

Puts the focus on the table.

Definition at line 116 of file PointFileController.java.

setUpView()

```
void ui.PointFileController.setUpView ( )
```

Definition at line 136 of file PointFileController.java.

updateModel()

```
void ui.PointFileController.updateModel ( )
```

Definition at line 121 of file PointFileController.java.

updateView()

```
void ui.PointFileController.updateView ( )
```

Definition at line 128 of file PointFileController.java.

6.77.4 Member Data Documentation

viewLabel

```
JLabel ui.PointFileController.viewLabel [package]
```

The label of the view.

Definition at line 98 of file PointFileController.java.

viewTable

`JTable ui.PointFileController.viewTable [package]`

The table of the view.

Definition at line 93 of file `PointFileController.java`.

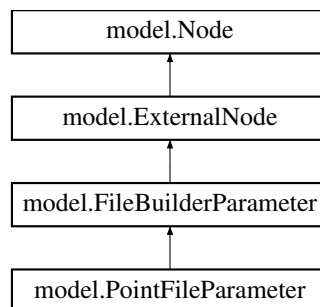
The documentation for this class was generated from the following file:

- [src/ui/PointFileController.java](#)

6.78 model.PointFileParameter Class Reference

A parameter used to build a point file.

Inheritance diagram for `model.PointFileParameter`:

**Public Member Functions**

- [PointFileParameter](#) (String [name](#), String [fileName](#), String [pointTag](#), String [pointFileTag](#), String [pointTagValue](#))
- boolean [fromFile](#) (File file) throws IOException
- boolean [isValid](#) ()
- boolean [isValidTable](#) ()
- [NodeController](#) [setUpController](#) ()
- String [getFileContent](#) ()
- List< String > [getListX](#) ()
- List< String > [getListY](#) ()

Static Protected Member Functions

- static String [getTaggedValue](#) (String [tag](#), File file)

Protected Attributes

- List< String > [listX](#)
The list of x values to be written in the file.
- List< String > [listY](#)
The list of y values to be written in the file.
- String [pointTag](#)
The tag used for the point parameter.
- String [pointFileTag](#)
- String [pointTagValue](#)
The value of the point parameter indicating that point is read from a file.

Additional Inherited Members

6.78.1 Detailed Description

A parameter used to build a point file.

Definition at line 75 of file PointFileParameter.java.

6.78.2 Constructor & Destructor Documentation

PointFileParameter()

```
model.PointFileParameter.PointFileParameter (
    String name,
    String fileName,
    String pointTag,
    String pointFileTag,
    String pointTagValue )
```

Construct a point file builder parameter.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file to be written
<i>pointTag</i>	the tag used for the point parameter
<i>pointFileTag</i>	the tag used for the point file parameter
<i>pointTagValue</i>	the value of the rain parameter indicating that point is read from a file

Definition at line 120 of file PointFileParameter.java.

6.78.3 Member Function Documentation

fromFile()

```
boolean model.PointFileParameter.fromFile (
    File file ) throws IOException
```

If the parameters.txt file indicates that point is read from a file, this method will attempt to initialize this node values from that point file.

Definition at line 137 of file PointFileParameter.java.

getFileContent()

```
String model.PointFileParameter.getFileContent ( )
```

Definition at line 268 of file PointFileParameter.java.

getListX()

```
List<String> model.PointFileParameter.getListX ( )
```

Returns

the list x

Definition at line 283 of file PointFileParameter.java.

getListY()

```
List<String> model.PointFileParameter.getListY ( )
```

Returns

the list y

Definition at line 290 of file PointFileParameter.java.

getTaggedValue()

```
static String model.PointFileParameter.getTaggedValue (
    String tag,
    File file ) [static], [protected]
```

Returns the value associated with the tag in the specified parameters.txt file.

Parameters

<i>tag</i>	the tag to look for
<i>file</i>	the parameters.txt file to read

Returns

the value associated with the tag if it exists or an empty string otherwise.

Definition at line 186 of file PointFileParameter.java.

isValid()

```
boolean model.PointFileParameter.isValid ( )
```

Returns true if the list of values are valid. Both lists must be the same length. Each value in this list greater than its predecessor. The point list must contain only non negative numbers. The table must be Enabled

Returns

true if the list of values are valid.

Definition at line 220 of file PointFileParameter.java.

isValidTable()

```
boolean model.PointFileParameter.isValidTable ( )
```

Returns true if the list of values are valid. Both lists must be the same length. Each value in this list greater than its predecessor. The point list must contain only non negative numbers. The table can be enabled or disabled

Returns

true if the list of values are valid.

Definition at line 236 of file PointFileParameter.java.

setUpController()

`NodeController` `model.PointFileParameter.setUpController ()`

Definition at line 263 of file PointFileParameter.java.

6.78.4 Member Data Documentation

listX

`List<String>` `model.PointFileParameter.listX` [protected]

The list of x values to be written in the file.

Definition at line 79 of file PointFileParameter.java.

listY

`List<String>` `model.PointFileParameter.listY` [protected]

The list of y values to be written in the file.

Definition at line 84 of file PointFileParameter.java.

pointFileTag

`String` `model.PointFileParameter.pointFileTag` [protected]

- The tag used for the point file parameter. Needed to initialize this parameter

Definition at line 95 of file PointFileParameter.java.

pointTag

`String` `model.PointFileParameter.pointTag` [protected]

The tag used for the point parameter.

Needed to initialize this parameter

Definition at line 89 of file PointFileParameter.java.

pointTagValue

`String` `model.PointFileParameter.pointTagValue` [protected]

The value of the point parameter indicating that point is read from a file.

Needed to initialize this parameter

Definition at line 101 of file PointFileParameter.java.

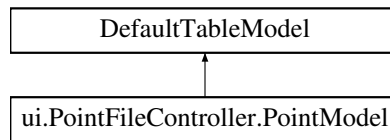
The documentation for this class was generated from the following file:

- `src/model/PointFileParameter.java`

6.79 ui.PointFileController.PointModel Class Reference

The model used by the view table.

Inheritance diagram for ui.PointFileController.PointModel:



Public Member Functions

- Class<?> [getColumnClass](#) (int columnIndex)
- int [getColumnCount](#) ()
- String [getColumnName](#) (int columnIndex)
- int [getRowCount](#) ()
- Object [getValueAt](#) (int rowIndex, int columnIndex)
- boolean [isCellEditable](#) (int rowIndex, int columnIndex)
- void [setValueAt](#) (Object aValue, int rowIndex, int columnIndex)
- boolean [isValidCell](#) (int row, int column)

Package Functions

- [PointModel](#) ()

Package Attributes

- java.util.List< String > [listX](#)
The list of x values.
- List< String > [listY](#)
The list of y value.

6.79.1 Detailed Description

The model used by the view table.

This model is an extension of the DefaultTableModel directly used by the JTable. It uses the model defined by PointFileParameter to get its values.

Definition at line 239 of file PointFileController.java.

6.79.2 Constructor & Destructor Documentation

PointModel()

```
ui.PointFileController.PointModel.PointModel ( ) [package]
```

Constructs a [PointModel](#) for a JTable using the values stored in the RainFileParameter.

Definition at line 256 of file PointFileController.java.

6.79.3 Member Function Documentation

getColumnClass()

```
Class<?> ui.PointFileController.PointModel.getColumnClass (
    int columnIndex )
```

Returns the most specific superclass for all the cell values in the column, in this case String.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the common ancestor class of the object values in the model.

Definition at line 272 of file PointFileController.java.

getColumnCount()

```
int ui.PointFileController.PointModel.getColumnCount ( )
```

Returns

the number of columns in the model.

Definition at line 280 of file PointFileController.java.

getColumnName()

```
String ui.PointFileController.PointModel.getColumnName (
    int columnIndex )
```

Returns the name of the column at columnIndex.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the name of the column.

Definition at line 293 of file PointFileController.java.

getRowCount()

```
int ui.PointFileController.PointModel.getRowCount ( )
```

Returns the number of rows in the model. That number includes an empty last line where the user can add new input.

Returns

the number of rows in the model.

Definition at line 312 of file PointFileController.java.

getValueAt()

```
Object ui.PointFileController.PointModel.getValueAt (
    int rowIndex,
    int columnIndex )
```

Parameters

<i>rowIndex</i>	the row whose value is to be queried
<i>columnIndex</i>	the column whose value is to be queried

Returns

the value for the cell at columnIndex and rowIndex.

Definition at line 327 of file PointFileController.java.

isCellEditable()

```
boolean ui.PointFileController.PointModel.isCellEditable (
    int rowIndex,
    int columnIndex )
```

Returns true if the cell at rowIndex and columnIndex is editable, which is the case except for cell (0,0)

Parameters

<i>rowIndex</i>	the row whose value to be queried
<i>columnIndex</i>	the column whose value to be queried

Returns

true if the cell at rowIndex and columnIndex is editable.

Definition at line 348 of file PointFileController.java.

isValidCell()

```
boolean ui.PointFileController.PointModel.isValidCell (
    int row,
    int column )
```

Parameters

<i>row</i>	the row of the cell
<i>column</i>	the column of the cell

Returns

true if the value of the cell is valid.

Definition at line 395 of file PointFileController.java.

setValueAt()

```
void ui.PointFileController.PointModel.setValueAt (
    Object aValue,
    int rowIndex,
    int columnIndex )
```

Sets the value in the cell at columnIndex and rowIndex to aValue. This method also sets the value in the user model.

Parameters

<i>aValue</i>	the new value
<i>rowIndex</i>	the row whose value is to be changed
<i>columnIndex</i>	the column whose value is to be changed

Definition at line 366 of file PointFileController.java.

6.79.4 Member Data Documentation**listX**

```
java.util.List<String> ui.PointFileController.PointModel.listX [package]
```

The list of x values.

Definition at line 244 of file PointFileController.java.

listY

```
List<String> ui.PointFileController.PointModel.listY [package]
```

The list of y value.

Definition at line 249 of file PointFileController.java.

The documentation for this class was generated from the following file:

- [src/ui/PointFileController.java](#)

6.80 model.MultipleChoiceParameter.PossibleValue Class Reference

A possible value is constituted of two strings.

Public Member Functions

- String [getValue](#) ()
- String [getPrintedValue](#) ()
- String [toString](#) ()

Package Functions

- [PossibleValue](#) (String name, String value)
- [PossibleValue](#) (String name, String value, String printedValue)

6.80.1 Detailed Description

A possible value is constituted of two strings.

One is used for display purposes and the other for storage, most notably on the generated parameters.txt file.

Definition at line 222 of file MultipleChoiceParameter.java.

6.80.2 Constructor & Destructor Documentation

PossibleValue() [1/2]

```
model.MultipleChoiceParameter.PossibleValue.PossibleValue (
    String name,
    String value ) [package]
```

Constructs a possible value with the provided name and actual storage value.

Parameters

<i>name</i>	the display name of the value
<i>value</i>	the string used for storage of the value

Definition at line 249 of file MultipleChoiceParameter.java.

PossibleValue() [2/2]

```
model.MultipleChoiceParameter.PossibleValue.PossibleValue (
    String name,
    String value,
    String printedValue ) [package]
```

Constructs a possible value with the provided name and actual storage value, and a different printed value.

Parameters

<i>name</i>	the display name of the value
<i>value</i>	the string used for storage of the value
<i>printedValue</i>	the string used for printing in the parameters.txt file

Definition at line 267 of file MultipleChoiceParameter.java.

6.80.3 Member Function Documentation

getPrintedValue()

```
String model.MultipleChoiceParameter.PossibleValue.getPrintedValue ( )
```

Returns

the string used for storage of the value.

Definition at line 283 of file MultipleChoiceParameter.java.

getValue()

```
String model.MultipleChoiceParameter.PossibleValue.getValue ( )
```

Returns

the string printed in the parameters.txt file.

Definition at line 276 of file MultipleChoiceParameter.java.

toString()

```
String model.MultipleChoiceParameter.PossibleValue.toString ( )
```

Returns

the display name of the value.

Definition at line 291 of file MultipleChoiceParameter.java.

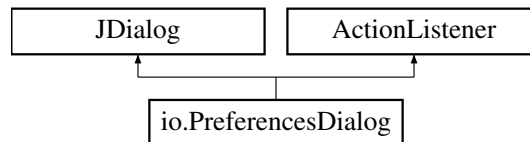
The documentation for this class was generated from the following file:

- src/model/[MultipleChoiceParameter.java](#)

6.81 io.PreferencesDialog Class Reference

An instance of this class is a JDialog corresponding to the preferences window of the user interface.

Inheritance diagram for io.PreferencesDialog:

**Public Member Functions**

- [PreferencesDialog](#) ()
- void [actionPerformed](#) (ActionEvent evt)

6.81.1 Detailed Description

An instance of this class is a JDialog corresponding to the preferences window of the user interface.

The preferences are saved in a file named settings.properties, which is located in the hidden directory ./fullswof_ui in the user directory.

See also

javax.swing.JDialog

Definition at line 114 of file PreferencesDialog.java.

6.81.2 Constructor & Destructor Documentation

PreferencesDialog()

```
io.PreferencesDialog.PreferencesDialog ( )
```

Creates and displays a preferences setting window

Definition at line 216 of file PreferencesDialog.java.

6.81.3 Member Function Documentation**actionPerformed()**

```
void io.PreferencesDialog.actionPerformed (
    ActionEvent evt )
```

Called when a 'Browse', 'Apply' or 'Cancel' button is clicked. The resulting action depends on the command associated with the button.

Definition at line 247 of file PreferencesDialog.java.

The documentation for this class was generated from the following file:

- [src/io/PreferencesDialog.java](#)

6.82 io.Procedures Class Reference

This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project.

Static Public Member Functions

- static void [about](#) ()
- static void [close](#) ()
- static String [getPathRedirection](#) ()
- static String [getPathRedirectionBrowse](#) ()
- static [Node](#) [] [getAvailableConfigurations](#) ()
- static [Node](#) [getDefaultConfiguration](#) ()
- static File [getLastDirectoryUsed](#) ()
- static String [] [getLaunchingCommands](#) ()
- static ResourceBundle [getMessages](#) ()
- static File [getOutputDirectory](#) ()
- static List< File > [getRecentFilesList](#) ()
- static [NodeController](#) [getWorkingController](#) ()
- static File [getWorkingDirectory](#) ()
- static [Node](#) [getWorkingModel](#) ()
- static File [getWorkingProject](#) ()
- static [MainFrame](#) [getWorkingUI](#) ()
- static boolean [hasChangedSinceLastSave](#) ()
- static void [help](#) ()
- static void [importParameters](#) ()
- static void [newProject](#) ()
- static void [newProjectAfterConfirmation](#) ()
- static void [openProject](#) ()
- static boolean [openProject](#) (File f)
- static void [openVisualisation](#) ()

- static boolean `projectIsReady` ()
- static void `runProject` ()
- static void `saveProject` ()
- static void `saveProjectAs` ()
- static void `setPathRedirection` (String pathRedirection)
- static void `setPathRedirectionBrowse` (String pathRedirectionBrowse)
- static void `setDefaultConfiguration` (Node configuration)
- static void `setLastDirectoryUsed` (File lastFileUsed)
- static void `setLaunchingCommands` (String[] launchingCommands)
- static void `setModel` (Node model)
- static void `setOutputDirectory` (File outputDirectory)
- static void `setPreferences` ()
- static void `setRecentFilesList` (List< File > recentFilesList)
- static void `setVerboseParametersFiles` (boolean verboseParametersFiles)
- static boolean `isFile` (String nameFile)

Check if the file exist.

- static File `getFile` (String nameFile)

the file

Static Public Attributes

- static ResourceBundle `messages`

The resource bundle containing the locale-specific strings displayed on the user interface.

- static Node `workingModel`

The current FullSWOF configuration being used by the application.

Static Package Attributes

- static boolean `changeSinceLastSave`

True if the parameters have been changed since the last time the project was saved.

6.82.1 Detailed Description

This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project.

Definition at line 92 of file Procedures.java.

6.82.2 Member Function Documentation

`about()`

```
static void io.Procedures.about ( ) [static]
```

Displays credits and license information.

Definition at line 211 of file Procedures.java.

close()

```
static void io.Procedures.close ( ) [static]
```

Terminates the application after a confirmation message.

Definition at line 220 of file Procedures.java.

getAvailableConfigurations()

```
static Node [] io.Procedures.getAvailableConfigurations ( ) [static]
```

Returns

an array of FullSWOF configurations available for projects

Definition at line 245 of file Procedures.java.

getDefaultConfiguration()

```
static Node io.Procedures.getDefaultConfiguration ( ) [static]
```

Returns

the configuration used for new projects

Definition at line 252 of file Procedures.java.

getFile()

```
static File io.Procedures.getFile (
    String nameFile ) [static]
```

the file

Parameters

<i>nameFile</i>	name file
-----------------	-----------

Returns

the file

Definition at line 1136 of file Procedures.java.

getLastDirectoryUsed()

```
static File io.Procedures.getLastDirectoryUsed ( ) [static]
```

Returns

the last directory opened with a file chooser

Definition at line 259 of file Procedures.java.

getLaunchingCommands()

```
static String [] io.Procedures.getLaunchingCommands ( ) [static]
```

Returns

an array of commands used to launch each version of the FullSWOF software

Definition at line 270 of file Procedures.java.

getMessages()

```
static ResourceBundle io.Procedures.getMessages ( ) [static]
```

Returns

the resource bundle used by the application to display UI messages

Definition at line 278 of file Procedures.java.

getOutputDirectory()

```
static File io.Procedures.getOutputDirectory ( ) [static]
```

Returns

the output directory used when FullSWOF is launched

Definition at line 285 of file Procedures.java.

getPathRedirection()

```
static String io.Procedures.getPathRedirection ( ) [static]
```

Returns

The directory opened with a file chooser

Definition at line 232 of file Procedures.java.

getPathRedirectionBrowse()

```
static String io.Procedures.getPathRedirectionBrowse ( ) [static]
```

Returns

The directory opened with a file chooser (button Browse)

Definition at line 239 of file Procedures.java.

getRecentFilesList()

```
static List<File> io.Procedures.getRecentFilesList ( ) [static]
```

Returns

the list of recently used files

Definition at line 292 of file Procedures.java.

getWorkingController()

```
static NodeController io.Procedures.getWorkingController ( ) [static]
```

Returns

the controller associated with the current configuration

Definition at line 299 of file Procedures.java.

getWorkingDirectory()

```
static File io.Procedures.getWorkingDirectory ( ) [static]
```

Returns

the directory associated the working project

Definition at line 306 of file Procedures.java.

getWorkingModel()

```
static Node io.Procedures.getWorkingModel ( ) [static]
```

Returns

the current FullSWOF configuration being used by the application

Definition at line 313 of file Procedures.java.

getWorkingProject()

```
static File io.Procedures.getWorkingProject ( ) [static]
```

Returns

the project file (*.fsp file) currently opened in the application

Definition at line 320 of file Procedures.java.

getWorkingUI()

```
static MainFrame io.Procedures.getWorkingUI ( ) [static]
```

Returns

the main window currently used by the interface

Definition at line 327 of file Procedures.java.

hasChangedSinceLastSave()

```
static boolean io.Procedures.hasChangedSinceLastSave ( ) [static]
```

Returns

true if the parameters have been changed since the last time the project was saved

Definition at line 335 of file Procedures.java.

help()

```
static void io.Procedures.help ( ) [static]
```

Opens a HTML help file in a browser. The file chosen depends on the current locale. The method will try to get the file corresponding to the current localization or fall back on a default file. The method to open a web browser is OS-dependent.

Definition at line 346 of file Procedures.java.

importParameters()

```
static void io.Procedures.importParameters ( ) [static]
```

Allows the user to chose a file and attempt to read it as a parameters file to import its parameters in the working project. There is no restriction on the name or format of the file, but it should follow the rules of a parameters.txt file or the import will not succeed. The method does not check the validity of the file, the parameters are simply left unmodified if the syntax of the file is bad.

Definition at line 376 of file Procedures.java.

isFile()

```
static boolean io.Procedures.isFile (
    String nameFile ) [static]
```

Check if the file exist.

Parameters

<i>nameFile</i>	name file
-----------------	-----------

Returns

True : the file exist

Definition at line 1117 of file Procedures.java.

newProject()

```
static void io.Procedures.newProject ( ) [static]
```

Allows the user to create a new project, using the default configuration choice. This method is called upon starting the application.

Definition at line 412 of file Procedures.java.

newProjectAfterConfirmation()

```
static void io.Procedures.newProjectAfterConfirmation ( ) [static]
```

Same as [newProject\(\)](#), after a confirmation message displayed if the current project has been changed and not saved.

Definition at line 430 of file Procedures.java.

openProject() [1/2]

```
static void io.Procedures.openProject ( ) [static]
```

Allows the user to open a project created with FullSWOF_UI (*.fsp) file. A confirmation message is displayed if the current project has been changed and not saved.

Definition at line 447 of file Procedures.java.

openProject() [2/2]

```
static boolean io.Procedures.openProject (
    File f ) [static]
```

Attempts to open the project save in a file

Parameters

<i>f</i>	the project file
----------	------------------

Returns

true if the project was successfully opened

Definition at line 471 of file Procedures.java.

openVisualisation()

```
static void io.Procedures.openVisualisation ( ) [static]
```

Allows the user to open a previously computed result file (*.dat) to get a 3D visualization of it.

Definition at line 542 of file Procedures.java.

projectIsReady()

```
static boolean io.Procedures.projectIsReady ( ) [static]
```

Called upon saving or opening of a project to check that all the necessary directories and files are present on the disk

Returns

true if all the necessary directories and files are present on the disk, false otherwise

Definition at line 570 of file Procedures.java.

runProject()

```
static void io.Procedures.runProject ( ) [static]
```

Allows the user to launch FullSWOF with the parameters of the current project. The parameters must be valid, otherwise an error message is displayed. If the project has not been saved, the interface will ask the user to do it.

Definition at line 620 of file Procedures.java.

saveProject()

```
static void io.Procedures.saveProject ( ) [static]
```

Saves the current project, overwriting the old files if the user allows it. If the current project is null (first time it is saved), this method works in the same way as [saveProjectAs\(\)](#) .

Definition at line 684 of file Procedures.java.

saveProjectAs()

```
static void io.Procedures.saveProjectAs ( ) [static]
```

Allows the user to save the current project as a new file. This new file is the current project after this operation.

Definition at line 783 of file Procedures.java.

setDefaultConfiguration()

```
static void io.Procedures.setDefaultConfiguration (
    Node configuration ) [static]
```

Sets the configuration used for new projects.

Parameters

<i>configuration</i>	the configuration to be used for new projects
----------------------	---

Definition at line 863 of file Procedures.java.

setLastDirectoryUsed()

```
static void io.Procedures.setLastDirectoryUsed (
    File lastFileUsed ) [static]
```

Sets the last directory used. The next time a file chooser is used, it will be opened at this location.

Parameters

<i>lastFileUsed</i>	the last file used. This can be a directory or a file, in which case the parent directory of this file will be considered the last directory used.
---------------------	--

Definition at line 877 of file Procedures.java.

setLaunchingCommands()

```
static void io.Procedures.setLaunchingCommands (
    String [] launchingCommands ) [static]
```

Sets the list of commands used to launch each version of the FullSWOF software. This array should have the same length as availableConfigurations, as each *i*th command corresponds to the *i*th configuration.

Parameters

<i>launchingCommands</i>	the list of commands used to launch each version of the FullSWOF software
--------------------------	---

Exceptions

<i>IllegalStateException</i>	if the array passed is not the same length as availableConfigurations
------------------------------	---

Definition at line 900 of file Procedures.java.

setModel()

```
static void io.Procedures.setModel (
    Node model ) [static]
```

Changes the model (FullSWOF configuration) used by the interface. All parameters are reset and the interface is updated with the new view. This method is called only upon creation of a new project.

Parameters

<i>model</i>	the new model to be set
--------------	-------------------------

Definition at line 916 of file Procedures.java.

setOutputDirectory()

```
static void io.Procedures.setOutputDirectory (
    File outputDirectory ) [static]
```

Sets the output directory used when FullSWOF is launched.

Parameters

<i>outputDirectory</i>	the output directory used when FullSWOF is launched
------------------------	---

Definition at line 934 of file Procedures.java.

setPathRedirection()

```
static void io.Procedures.setPathRedirection (
    String pathRedirection ) [static]
```

Parameters

<i>pathRedirection</i>	The new directory opened with a file chooser
------------------------	--

Definition at line 844 of file Procedures.java.

setPathRedirectionBrowse()

```
static void io.Procedures.setPathRedirectionBrowse (
    String pathRedirectionBrowse ) [static]
```

Parameters

<i>pathRedirectionBrowse</i>	The new directory opened with a file chooser (button Browse)
------------------------------	--

Definition at line 852 of file Procedures.java.

setPreferences()

```
static void io.Procedures.setPreferences ( ) [static]
```

Opens the preferences dialog.

See also

[PreferencesDialog](#)

Definition at line 944 of file Procedures.java.

setRecentFilesList()

```
static void io.Procedures.setRecentFilesList (
    List< File > recentFilesList ) [static]
```

Sets the list of recently used project files.

Parameters

<i>recentFilesList</i>	the list of recently used project files
------------------------	---

Definition at line 955 of file Procedures.java.

setVerboseParametersFiles()

```
static void io.Procedures.setVerboseParametersFiles (
    boolean verboseParametersFiles ) [static]
```

Sets the option for generating verbose parameters file.

Parameters

<i>verboseParametersFiles</i>	true if the generated parameters.txt must contain descriptive comments for each parameter, false if it limited to the tags and their values
-------------------------------	---

Definition at line 968 of file Procedures.java.

6.82.3 Member Data Documentation

changeSinceLastSave

```
boolean io.Procedures.changeSinceLastSave [static], [package]
```

True if the parameters have been changed since the last time the project was saved.

FullSWOF_UI does not keep an undo history. This variable is used only to decide if confirmation messages should be displayed when the user attempts an action such as opening another project without saving the current project first

Definition at line 194 of file Procedures.java.

messages

```
ResourceBundle io.Procedures.messages [static]
```

The resource bundle containing the locale-specific strings displayed on the user interface.

Definition at line 106 of file Procedures.java.

workingModel

Node `io.Procedures.workingModel` [static]

The current FullSWOF configuration being used by the application.

Definition at line 150 of file Procedures.java.

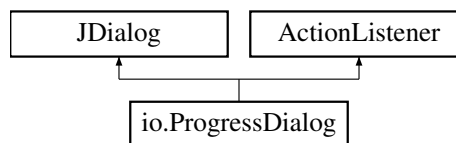
The documentation for this class was generated from the following file:

- `src/io/Procedures.java`

6.83 io.ProgressDialog Class Reference

A dialog box including a progress bar, a console display and a cancel button.

Inheritance diagram for `io.ProgressDialog`:

**Public Member Functions**

- [ProgressDialog](#) (Thread interruptableThread)
- void [actionPerformed](#) (ActionEvent e)
- void [addConsoleLine](#) (String line)
- void [updateProgress](#) (int progress)

6.83.1 Detailed Description

A dialog box including a progress bar, a console display and a cancel button.

This dialog box is linked to a thread that can be interrupted by pressing the button.

Definition at line 81 of file ProgressDialog.java.

6.83.2 Constructor & Destructor Documentation**ProgressDialog()**

```
io.ProgressDialog.ProgressDialog (
    Thread interruptableThread )
```

Constructs and displays the dialog box

Parameters

<code>interruptableThread</code>	the thread that can be interrupted by pressing the cancel button
----------------------------------	--

Definition at line 117 of file ProgressDialog.java.

6.83.3 Member Function Documentation

actionPerformed()

```
void io.ProgressDialog.actionPerformed (
    ActionEvent e )
```

Action performed when the user presses the cancel button
Definition at line 146 of file ProgressDialog.java.

addConsoleLine()

```
void io.ProgressDialog.addConsoleLine (
    String line )
```

Appends a line to the console

Parameters

<i>line</i>	the line to append
-------------	--------------------

Definition at line 158 of file ProgressDialog.java.

updateProgress()

```
void io.ProgressDialog.updateProgress (
    int progress )
```

Updates the progress bar

Parameters

<i>progress</i>	an integer between 0 and 100
-----------------	------------------------------

Definition at line 174 of file ProgressDialog.java.

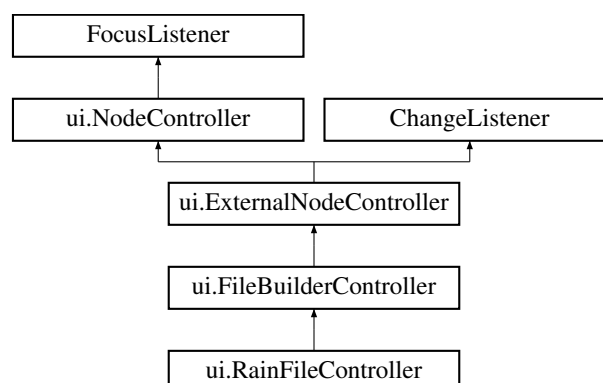
The documentation for this class was generated from the following file:

- [src/io/ProgressDialog.java](#)

6.84 ui.RainFileController Class Reference

A controller for a rain file builder node.

Inheritance diagram for ui.RainFileController:



Classes

- class [CellRenderer](#)
An instance of this class is used to render the cells of the table in the view.
- class [RainModel](#)
The model used by the view table.

Public Member Functions

- [RainFileController](#) ([RainFileParameter](#) model)
- void [highlightView](#) ()
- void [updateModel](#) ()
- void [updateView](#) ()
- void [setUpView](#) ()

Package Attributes

- JTable [viewTable](#)
The table of the view.
- JLabel [viewLabel](#)
The label of the view.

Additional Inherited Members

6.84.1 Detailed Description

A controller for a rain file builder node.

This controller can set up a view that includes an editable table where the user can write time and rain value.
Definition at line 87 of file RainFileController.java.

6.84.2 Constructor & Destructor Documentation

RainFileController()

```
ui.RainFileController.RainFileController (
    RainFileParameter model )
```

Constructs a controller for a RainFileParameter.

Parameters

<i>model</i>	the node to be controlled
--------------	---------------------------

Definition at line 117 of file RainFileController.java.

6.84.3 Member Function Documentation

highlightView()

```
void ui.RainFileController.highlightView ( )
```

Puts the focus on the table.

Definition at line 126 of file RainFileController.java.

setUpView()

```
void ui.RainFileController.setUpView ( )
```

Definition at line 146 of file RainFileController.java.

updateModel()

```
void ui.RainFileController.updateModel ( )
```

Definition at line 131 of file RainFileController.java.

updateView()

```
void ui.RainFileController.updateView ( )
```

Definition at line 138 of file RainFileController.java.

6.84.4 Member Data Documentation**viewLabel**

```
JLabel ui.RainFileController.viewLabel [package]
```

The label of the view.

Definition at line 108 of file RainFileController.java.

viewTable

```
JTable ui.RainFileController.viewTable [package]
```

The table of the view.

Definition at line 103 of file RainFileController.java.

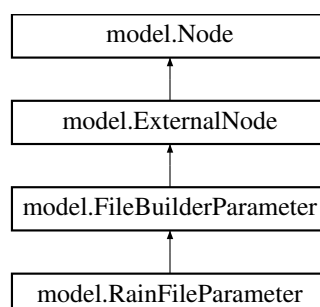
The documentation for this class was generated from the following file:

- [src/ui/RainFileController.java](#)

6.85 model.RainFileParameter Class Reference

A parameter used to build a rain file.

Inheritance diagram for model.RainFileParameter:



Public Member Functions

- `RainFileParameter` (String `name`, String `fileName`, String `rainTag`, String `rainFileTag`, String `rainTagValue`)
- boolean `fromFile` (File file) throws IOException
- boolean `isValid` ()
- boolean `isValidTable` ()
- void `setTime` (List< String > `time`)
- void `setRain` (List< String > `rain`)
- List< String > `getTime` ()
- List< String > `getRain` ()
- `NodeController` `setUpController` ()
- String `getFileContent` ()

Static Protected Member Functions

- static String `getTaggedValue` (String `tag`, File file)

Protected Attributes

- List< String > `time`
The list of time values to be written in the file.
- List< String > `rain`
The list of rain values to be written in the file.
- String `rainTag`
The tag used for the rain parameter.
- String `rainFileTag`
- String `rainTagValue`
The value of the rain parameter indicating that rain is read from a file.

Additional Inherited Members

6.85.1 Detailed Description

A parameter used to build a rain file.

Definition at line 76 of file RainFileParameter.java.

6.85.2 Constructor & Destructor Documentation

RainFileParameter()

```
model.RainFileParameter.RainFileParameter (
    String name,
    String fileName,
    String rainTag,
    String rainFileTag,
    String rainTagValue )
```

Construct a rain file builder parameter.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file to be written

Parameters

<i>rainTag</i>	the tag used for the rain parameter
<i>rainFileTag</i>	the tag used for the rain file parameter
<i>rainTagValue</i>	the value of the rain parameter indicating that rain is read from a file

Definition at line 121 of file RainFileParameter.java.

6.85.3 Member Function Documentation**fromFile()**

```
boolean model.RainFileParameter.fromFile (
    File file ) throws IOException
```

If the parameters.txt file indicates that rain is read from a file, this method will attempt to initialize this node values from that rain file.

Definition at line 137 of file RainFileParameter.java.

getFileContent()

```
String model.RainFileParameter.getFileContent ( )
```

Definition at line 311 of file RainFileParameter.java.

getRain()

```
List<String> model.RainFileParameter.getRain ( )
```

Returns

the list of rain values to be written in the file.

Definition at line 301 of file RainFileParameter.java.

getTaggedValue()

```
static String model.RainFileParameter.getTaggedValue (
    String tag,
    File file ) [static], [protected]
```

Returns the value associated with the tag in the specified parameters.txt file.

Parameters

<i>tag</i>	the tag to look for
<i>file</i>	the parameters.txt file to read

Returns

the value associated with the tag if it exists or an empty string otherwise.

Definition at line 187 of file RainFileParameter.java.

getTime()

```
List<String> model.RainFileParameter.getTime ( )
```

Returns

the list of time values to be written in the file.

Definition at line 294 of file RainFileParameter.java.

isValid()

```
boolean model.RainFileParameter.isValid ( )
```

Returns true if the list of values are valid. Both lists must be the same length. The first value of the time list must be zero, and each value in this list greater than its predecessor. The rain list must contain only non negative numbers. The table must be Enabled

Returns

true if the list of values are valid.

Definition at line 221 of file RainFileParameter.java.

isValidTable()

```
boolean model.RainFileParameter.isValidTable ( )
```

Returns true if the list of values are valid. Both lists must be the same length. The first value of the time list must be zero, and each value in this list greater than its predecessor. The rain list must contain only non negative numbers. The table can be enabled or disabled

Returns

true if the list of values are valid.

Definition at line 237 of file RainFileParameter.java.

setRain()

```
void model.RainFileParameter.setRain (
    List< String > rain )
```

Sets the list of rain values to be written in the file.

Parameters

<i>rain</i>	the list of rain values to be written in the file
-------------	---

Definition at line 286 of file RainFileParameter.java.

setTime()

```
void model.RainFileParameter.setTime (
    List< String > time )
```

Sets the list of time values to be written in the file.

Parameters

<i>time</i>	the list of time values to be written in the file
-------------	---

Definition at line 274 of file RainFileParameter.java.

setUpController()

`NodeController` `model.RainFileParameter.setUpController ()`

Definition at line 306 of file RainFileParameter.java.

6.85.4 Member Data Documentation**rain**

`List<String> model.RainFileParameter.rain [protected]`

The list of rain values to be written in the file.

Definition at line 86 of file RainFileParameter.java.

rainFileTag

`String model.RainFileParameter.rainFileTag [protected]`

- The tag used for the rain file parameter. Needed to initialize this parameter

Definition at line 97 of file RainFileParameter.java.

rainTag

`String model.RainFileParameter.rainTag [protected]`

The tag used for the rain parameter.

Needed to initialize this parameter

Definition at line 91 of file RainFileParameter.java.

rainTagValue

`String model.RainFileParameter.rainTagValue [protected]`

The value of the rain parameter indicating that rain is read from a file.

Needed to initialize this parameter

Definition at line 103 of file RainFileParameter.java.

time

`List<String> model.RainFileParameter.time [protected]`

The list of time values to be written in the file.

Definition at line 81 of file RainFileParameter.java.

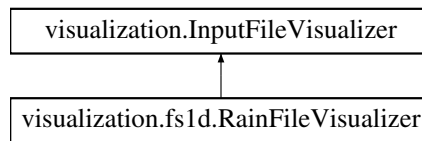
The documentation for this class was generated from the following file:

- `src/model/RainFileParameter.java`

6.86 visualization.fs1d.RainFileVisualizer Class Reference

A tool used to get a quick visualization of rain input files for FullSWOF_1D.

Inheritance diagram for visualization.fs1d.RainFileVisualizer:



Public Member Functions

- [Chart getVisualization](#) (File file) throws IOException

6.86.1 Detailed Description

A tool used to get a quick visualization of rain input files for FullSWOF_1D.

The file is represented as a chart showing the evolution of rain during time.

Definition at line 81 of file RainFileVisualizer.java.

6.86.2 Member Function Documentation

getVisualization()

[Chart](#) visualization.fs1d.RainFileVisualizer.getVisualization (
 File *file*) throws IOException

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).

Definition at line 83 of file RainFileVisualizer.java.

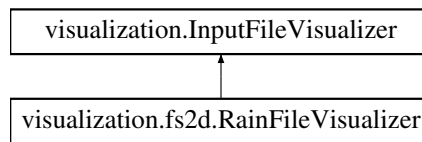
The documentation for this class was generated from the following file:

- src/visualization/fs1d/[RainFileVisualizer.java](#)

6.87 visualization.fs2d.RainFileVisualizer Class Reference

A tool used to get a quick visualization of rain input files for FullSWOF_2D.

Inheritance diagram for visualization.fs2d.RainFileVisualizer:



Public Member Functions

- [Chart getVisualization](#) (File file) throws IOException

6.87.1 Detailed Description

A tool used to get a quick visualization of rain input files for FullSWOF_2D.

The file is represented as a chart showing the evolution of rain during time.

Definition at line 80 of file RainFileVisualizer.java.

6.87.2 Member Function Documentation

getVisualization()

[Chart](#) visualization.fs2d.RainFileVisualizer.getVisualization (
 File *file*) throws IOException

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).

Definition at line 82 of file RainFileVisualizer.java.

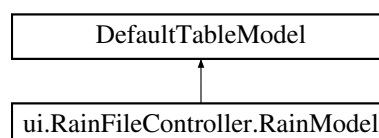
The documentation for this class was generated from the following file:

- src/visualization/fs2d/[RainFileVisualizer.java](#)

6.88 ui.RainFileController.RainModel Class Reference

The model used by the view table.

Inheritance diagram for ui.RainFileController.RainModel:



Public Member Functions

- `Class<?> getColumnClass` (int columnIndex)
- `int getColumnCount` ()
- `String getColumnName` (int columnIndex)
- `int getRowCount` ()
- `Object getValueAt` (int rowIndex, int columnIndex)
- `boolean isCellEditable` (int rowIndex, int columnIndex)
- `void setValueAt` (Object aValue, int rowIndex, int columnIndex)
- `boolean isValidCell` (int row, int column)

Package Functions

- `RainModel` ()

Package Attributes

- `List< String > time`
The list of time values.
- `List< String > rain`
The list of rain value.

6.88.1 Detailed Description

The model used by the view table.

This model is an extension of the DefaultTableModel directly used by the JTable. It uses the model defined by RainFileParameter to get its values.

Definition at line 249 of file RainFileController.java.

6.88.2 Constructor & Destructor Documentation

RainModel()

```
ui.RainFileController.RainModel.RainModel ( ) [package]
```

Constructs a `RainModel` for a JTable using the values stored in the RainFileParameter.

Definition at line 266 of file RainFileController.java.

6.88.3 Member Function Documentation

getColumnClass()

```
Class<?> ui.RainFileController.RainModel.getColumnClass (
    int columnIndex )
```

Returns the most specific superclass for all the cell values in the column, in this case String.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the common ancestor class of the object values in the model.

Definition at line 282 of file RainFileController.java.

getColumnCount()

```
int ui.RainFileController.RainModel.getColumnCount ( )
```

Returns

the number of columns in the model.

Definition at line 290 of file RainFileController.java.

getColumnName()

```
String ui.RainFileController.RainModel.getColumnName (
    int columnIndex )
```

Returns the name of the column at columnIndex.

Parameters

<i>columnIndex</i>	the index of the column
--------------------	-------------------------

Returns

the name of the column.

Definition at line 303 of file RainFileController.java.

getRowCount()

```
int ui.RainFileController.RainModel.getRowCount ( )
```

Returns the number of rows in the model. That number includes an empty last line where the user can add new input.

Returns

the number of rows in the model.

Definition at line 322 of file RainFileController.java.

getValueAt()

```
Object ui.RainFileController.RainModel.getValueAt (
    int rowIndex,
    int columnIndex )
```

Parameters

<i>rowIndex</i>	the row whose value is to be queried
<i>columnIndex</i>	the column whose value is to be queried

Returns

the value for the cell at columnIndex and rowIndex.

Definition at line 337 of file RainFileController.java.

isCellEditable()

```
boolean ui.RainFileController.RainModel.isCellEditable (
    int rowIndex,
    int columnIndex )
```

Returns true if the cell at rowIndex and columnIndex is editable, which is the case except for cell (0,0)

Parameters

<i>rowIndex</i>	the row whose value to be queried
<i>columnIndex</i>	the column whose value to be queried

Returns

true if the cell at rowIndex and columnIndex is editable.

Definition at line 358 of file RainFileController.java.

isValidCell()

```
boolean ui.RainFileController.RainModel.isValidCell (
    int row,
    int column )
```

Parameters

<i>row</i>	the row of the cell
<i>column</i>	the column of the cell

Returns

true if the value of the cell is valid.

Definition at line 405 of file RainFileController.java.

setValueAt()

```
void ui.RainFileController.RainModel.setValueAt (
    Object aValue,
    int rowIndex,
    int columnIndex )
```

Sets the value in the cell at columnIndex and rowIndex to aValue. This method also sets the value in the user model.

Parameters

<i>aValue</i>	the new value
---------------	---------------

Parameters

<i>rowIndex</i>	the row whose value is to be changed
<i>columnIndex</i>	the column whose value is to be changed

Definition at line 376 of file RainFileController.java.

6.88.4 Member Data Documentation**rain**

```
List<String> ui.RainFileController.RainModel.rain [package]
```

The list of rain value.

Definition at line 259 of file RainFileController.java.

time

```
List<String> ui.RainFileController.RainModel.time [package]
```

The list of time values.

Definition at line 254 of file RainFileController.java.

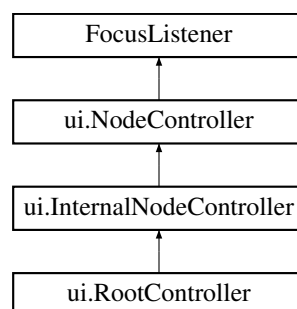
The documentation for this class was generated from the following file:

- [src/ui/RainFileController.java](#)

6.89 ui.RootController Class Reference

A controller for an internal node, especially suited for the root of the tree.

Inheritance diagram for ui.RootController:

**Public Member Functions**

- [RootController](#) ([InternalNode](#) node)
- boolean [validate](#) (File mainDirectory)

Package Functions

- void [setUpView](#) ()

Additional Inherited Members

6.89.1 Detailed Description

A controller for an internal node, especially suited for the root of the tree.

This class can still be used for other internal nodes, nor does the root need to use this controller. The view provided by this controller is simply better suited for the root. The view is a tabbed pane, where each child node is a tab.

See also

[ui.ParametersGroupController](#) for the other type of controller that can be used for internal nodes.

Definition at line 82 of file RootController.java.

6.89.2 Constructor & Destructor Documentation

RootController()

```
ui.RootController.RootController (
    InternalNode node )
```

Constructs a root controller for an internal node.

Parameters

<i>node</i>	the internal node to be controlled
-------------	------------------------------------

Definition at line 91 of file RootController.java.

6.89.3 Member Function Documentation

setUpView()

```
void ui.RootController.setUpView ( ) [package]
```

Instantiates the view for this controller. The view is a tabbed pane, where each child node is a tab.

Definition at line 123 of file RootController.java.

validate()

```
boolean ui.RootController.validate (
    File mainDirectory )
```

Applies validation procedures to the node. If one the child node is not validated, the corresponding tab is brought forward to help the user find the wrong value.

Definition at line 101 of file RootController.java.

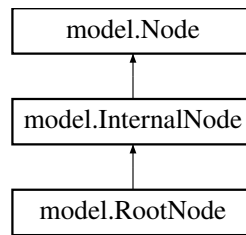
The documentation for this class was generated from the following file:

- [src/ui/RootController.java](#)

6.90 model.RootNode Class Reference

This class can be used for any internal node, but its controller is better suited to the root of the tree.

Inheritance diagram for model.RootNode:



Public Member Functions

- [RootNode](#) (String [name](#))
- [RootNode](#) (String [name](#), String [description](#))
- [NodeController](#) [setUpController](#) ()

Additional Inherited Members

6.90.1 Detailed Description

This class can be used for any internal node, but its controller is better suited to the root of the tree. The root controller offers a tabbed pane view, where each child node is a tab.

See also

[ui.RootController](#)

Definition at line 74 of file `RootNode.java`.

6.90.2 Constructor & Destructor Documentation

RootNode() [1/2]

```
model.RootNode.RootNode (
    String name )
```

Constructs a root node with the provided name.

Parameters

<i>name</i>	the name of the node
-------------	----------------------

Definition at line 83 of file `RootNode.java`.

RootNode() [2/2]

```
model.RootNode.RootNode (
    String name,
    String description )
```

Constructs a root node with the provided name and description.

Parameters

<i>name</i>	the name of the node
<i>description</i>	a description of the node

Definition at line 96 of file RootNode.java.

6.90.3 Member Function Documentation

setUpController()

`NodeController` `model.RootNode.setUpController ()`

Returns

a `ui.RootController` instance for this node.

See also

`ui.RootController`

Definition at line 105 of file RootNode.java.

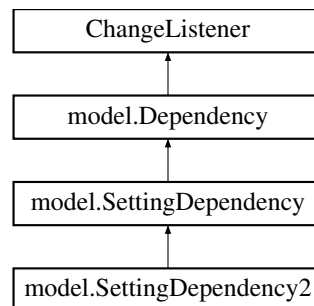
The documentation for this class was generated from the following file:

- `src/model/RootNode.java`

6.91 model.SettingDependency Class Reference

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Inheritance diagram for `model.SettingDependency`:



Public Member Functions

- `SettingDependency (ExternalNode master, ExternalNode slave, String targetValue)`
- `SettingDependency (ExternalNode master, ExternalNode slave, String targetValue, String slaveValue)`
- `String getSlaveValue ()`
- `boolean isRespected ()`
- `void resolve ()`

Protected Attributes

- `String slaveValue`

The value to which the slave node is to be set when the dependency is resolved.

Additional Inherited Members

6.91.1 Detailed Description

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Definition at line 66 of file SettingDependency.java.

6.91.2 Constructor & Destructor Documentation

SettingDependency() [1/2]

```
model.SettingDependency.SettingDependency (
    ExternalNode master,
    ExternalNode slave,
    String targetValue )
```

Constructs a setting dependency between the master and slave node for a target value. The slave node will be set to the target value when the dependency is resolved.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency

Definition at line 87 of file SettingDependency.java.

SettingDependency() [2/2]

```
model.SettingDependency.SettingDependency (
    ExternalNode master,
    ExternalNode slave,
    String targetValue,
    String slaveValue )
```

Constructs a setting dependency between the master and slave node for a target value. The slave node will be set to slaveValue when the dependency is resolved.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency
<i>slaveValue</i>	the value to which the slave node will be set upon resolution

Definition at line 108 of file SettingDependency.java.

6.91.3 Member Function Documentation

getSlaveValue()

```
String model.SettingDependency.getSlaveValue ( )
```

Returns

the value to which the slave node is to be set when the dependency is resolved.

Definition at line 118 of file SettingDependency.java.

isRespected()

```
boolean model.SettingDependency.isRespected ( )
```

Returns false if and only if the master node value is equal to the target value *and* the slave node value is different from the value specified by the dependency.

Returns

false if the master node value is equal to the target value and the slave node value is different from the value specified by the dependency.

Definition at line 132 of file SettingDependency.java.

resolve()

```
void model.SettingDependency.resolve ( )
```

Sets the slave node to the specified value if the master node is equal to the target value.

Definition at line 146 of file SettingDependency.java.

6.91.4 Member Data Documentation**slaveValue**

```
String model.SettingDependency.slaveValue [protected]
```

The value to which the slave node is to be set when the dependency is resolved.

Definition at line 72 of file SettingDependency.java.

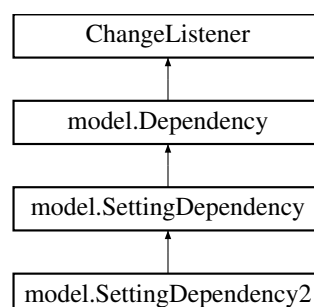
The documentation for this class was generated from the following file:

- src/model/[SettingDependency.java](#)

6.92 model.SettingDependency2 Class Reference

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Inheritance diagram for model.SettingDependency2:



Public Member Functions

- `SettingDependency2` (`ExternalNode` master, `ExternalNode` slave, `String` targetValue, `String` slaveValue, `ExternalNode` slaveGeneration)
- `void resolve` ()
- `void check` ()

Check if slaveGeneration is not valid and enabled Change the value of slave and master.

Additional Inherited Members

6.92.1 Detailed Description

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

If there are errors, it put old values

Definition at line 67 of file `SettingDependency2.java`.

6.92.2 Constructor & Destructor Documentation

SettingDependency2()

```
model.SettingDependency2.SettingDependency2 (
    ExternalNode master,
    ExternalNode slave,
    String targetValue,
    String slaveValue,
    ExternalNode slaveGeneration )
```

Constructs a setting dependency between the master and slave node for a target value. The slave node will be set to slaveValue when the dependency is resolved.

Parameters

<i>master</i>	the master node
<i>slave</i>	the slave node
<i>targetValue</i>	the triggering value of the dependency
<i>slaveValue</i>	the value to which the slave node will be set upon resolution
<i>slaveGeneration</i>	It is a node with the target b.

Definition at line 96 of file `SettingDependency2.java`.

6.92.3 Member Function Documentation

check()

```
void model.SettingDependency2.check ( )
```

Check if slaveGeneration is not valid and enabled Change the value of slave and master.

Definition at line 120 of file `SettingDependency2.java`.

resolve()

```
void model.SettingDependency2.resolve ( )
```

Sets the slave node to the specified value if the master node is equal to the target value.

Definition at line 107 of file SettingDependency2.java.

The documentation for this class was generated from the following file:

- src/model/[SettingDependency2.java](#)

6.93 io.Start Class Reference

The executable class used to launch the application.

Static Public Member Functions

- static void [main](#) (String[] args)

Static Public Attributes

- static Locale [currentLocale](#)

The locale used by the application.

6.93.1 Detailed Description

The executable class used to launch the application.

Definition at line 84 of file Start.java.

6.93.2 Member Function Documentation**main()**

```
static void io.Start.main (
    String [] args ) [static]
```

The executable method main

Parameters

<i>args</i>	not used
-------------	----------

Definition at line 98 of file Start.java.

6.93.3 Member Data Documentation**currentLocale**

```
Locale io.Start.currentLocale [static]
```

The locale used by the application.

Definition at line 89 of file Start.java.

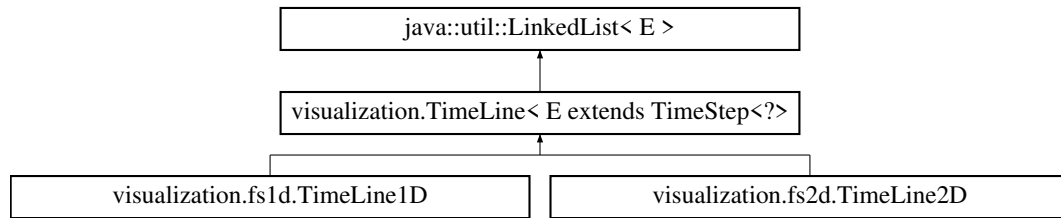
The documentation for this class was generated from the following file:

- src/io/[Start.java](#)

6.94 visualization.TimeLine< E extends TimeStep<?> Class Template Reference

A collection of time step in a FullSWOF evolution file.

Inheritance diagram for visualization.TimeLine< E extends TimeStep<?>:



Protected Member Functions

- [TimeLine](#) ()

6.94.1 Detailed Description

A collection of time step in a FullSWOF evolution file.

Parameters

<E>	the type time step in the collection
-----	--------------------------------------

Definition at line 71 of file TimeLine.java.

6.94.2 Constructor & Destructor Documentation

TimeLine()

`visualization.TimeLine< E extends TimeStep<?>.TimeLine () [protected]`

Constructs an empty time line.

Definition at line 77 of file TimeLine.java.

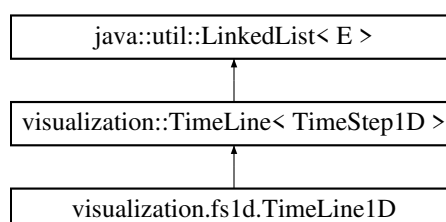
The documentation for this class was generated from the following file:

- src/visualization/[TimeLine.java](#)

6.95 visualization.fs1d.TimeLine1D Class Reference

A collection of [TimeStep1D](#) ordered by ascending time.

Inheritance diagram for visualization.fs1d.TimeLine1D:



Public Member Functions

- [TimeLine1D](#) ()
- boolean [add](#) ([TimeStep1D](#) t)
- [AnimatedChart](#) [getVisualisation](#) ()
- [ChartLine](#) [getXminCumulativeDischarge](#) ()
- [ChartLine](#) [getXminInstantDischarge](#) ()
- [ChartLine](#) [getXmaxCumulativeDischarge](#) ()
- [ChartLine](#) [getXmaxInstantDischarge](#) ()
- [ChartLine](#) [getXmaxWaterHeight](#) ()
- [ChartLine](#) [getXminWaterHeight](#) ()

Additional Inherited Members

6.95.1 Detailed Description

A collection of [TimeStep1D](#) ordered by ascending time.
Definition at line 76 of file [TimeLine1D.java](#).

6.95.2 Constructor & Destructor Documentation

[TimeLine1D\(\)](#)

`visualization.fs1d.TimeLine1D.TimeLine1D ()`
Constructs an empty timeline.
Definition at line 88 of file [TimeLine1D.java](#).

6.95.3 Member Function Documentation

[add\(\)](#)

`boolean visualization.fs1d.TimeLine1D.add (`
 [TimeStep1D](#) t `)`
Adds a time step in the timeline, conserving the timeline ordering.

Parameters

<i>t</i>	the time step to be added
----------	---------------------------

Definition at line 100 of file [TimeLine1D.java](#).

[getVisualisation\(\)](#)

[AnimatedChart](#) `visualization.fs1d.TimeLine1D.getVisualisation ()`
Returns an animated spatial visualization of the timeline. The values included in the representation are the ones indicated by the boolean variables in the class [visualization.VisualizationPane](#).

Returns

an animated chart.

Definition at line 119 of file TimeLine1D.java.

getXmaxCumulativeDischarge()

`ChartLine visualization.fs1d.TimeLine1D.getXmaxCumulativeDischarge ()`

Returns a chart line showing the cumulative discharge at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 174 of file TimeLine1D.java.

getXmaxInstantDischarge()

`ChartLine visualization.fs1d.TimeLine1D.getXmaxInstantDischarge ()`

Returns a chart line showing the instant discharge at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 198 of file TimeLine1D.java.

getXmaxWaterHeight()

`ChartLine visualization.fs1d.TimeLine1D.getXmaxWaterHeight ()`

Returns a chart line showing the evolution of the water height at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 215 of file TimeLine1D.java.

getXminCumulativeDischarge()

`ChartLine visualization.fs1d.TimeLine1D.getXminCumulativeDischarge ()`

Returns a chart line showing the cumulative discharge at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 133 of file TimeLine1D.java.

getXminInstantDischarge()

`ChartLine visualization.fs1d.TimeLine1D.getXminInstantDischarge ()`

Returns a chart line showing the instant discharge at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 157 of file TimeLine1D.java.

getXminWaterHeight()

`ChartLine visualization.fs1d.TimeLine1D.getXminWaterHeight ()`

Returns a chart line showing the evolution of the water height at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 233 of file TimeLine1D.java.

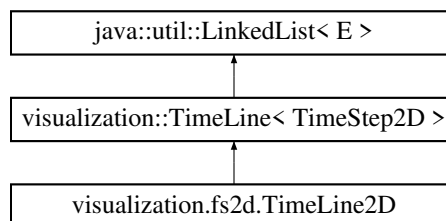
The documentation for this class was generated from the following file:

- `src/visualization/fs1d/TimeLine1D.java`

6.96 visualization.fs2d.TimeLine2D Class Reference

A collection of [TimeStep2D](#) ordered by ascending time.

Inheritance diagram for `visualization.fs2d.TimeLine2D`:

**Public Member Functions**

- [AnimatedScene](#) `getVisualisation ()`
- [ChartLine](#) `getXmaxCumulativeDischarge ()`
- [ChartLine](#) `getXmaxInstantDischarge ()`
- [ChartLine](#) `getXmaxWaterHeight ()`
- [ChartLine](#) `getXminCumulativeDischarge ()`
- [ChartLine](#) `getXminInstantDischarge ()`
- [ChartLine](#) `getXminWaterHeight ()`
- [ChartLine](#) `getYmaxCumulativeDischarge ()`
- [ChartLine](#) `getYmaxInstantDischarge ()`
- [ChartLine](#) `getYmaxWaterHeight ()`
- [ChartLine](#) `getYminCumulativeDischarge ()`
- [ChartLine](#) `getYminInstantDischarge ()`
- [ChartLine](#) `getYminWaterHeight ()`

Additional Inherited Members

6.96.1 Detailed Description

A collection of [TimeStep2D](#) ordered by ascending time.

Definition at line 76 of file TimeLine2D.java.

6.96.2 Member Function Documentation

getVisualisation()

```
AnimatedScene visualization.fs2d.TimeLine2D.getVisualisation ( )
```

Returns an animated spatial visualization of the timeline. The values included in the representation are the ones indicated by the boolean variables in the class [visualization.VisualizationPane](#).

Returns

an animated chart.

Definition at line 92 of file TimeLine2D.java.

getXmaxCumulativeDischarge()

```
ChartLine visualization.fs2d.TimeLine2D.getXmaxCumulativeDischarge ( )
```

Returns a chart line showing the cumulative discharge at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 107 of file TimeLine2D.java.

getXmaxInstantDischarge()

```
ChartLine visualization.fs2d.TimeLine2D.getXmaxInstantDischarge ( )
```

Returns a chart line showing the instant discharge at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 131 of file TimeLine2D.java.

getXmaxWaterHeight()

```
ChartLine visualization.fs2d.TimeLine2D.getXmaxWaterHeight ( )
```

Returns a chart line showing the water height at the right border during the length of the timeline.

Returns

a chart line.

Definition at line 148 of file TimeLine2D.java.

getXminCumulativeDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getXminCumulativeDischarge ()`

Returns a chart line showing the cumulative discharge at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 166 of file TimeLine2D.java.

getXminInstantDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getXminInstantDischarge ()`

Returns a chart line showing the instant discharge at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 190 of file TimeLine2D.java.

getXminWaterHeight()

`ChartLine visualization.fs2d.TimeLine2D.getXminWaterHeight ()`

Returns a chart line showing the water height at the left border during the length of the timeline.

Returns

a chart line.

Definition at line 207 of file TimeLine2D.java.

getYmaxCumulativeDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getYmaxCumulativeDischarge ()`

Returns a chart line showing the cumulative discharge at the top border during the length of the timeline.

Returns

a chart line.

Definition at line 225 of file TimeLine2D.java.

getYmaxInstantDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getYmaxInstantDischarge ()`

Returns a chart line showing the instant discharge at the top border during the length of the timeline.

Returns

a chart line.

Definition at line 249 of file TimeLine2D.java.

getYmaxWaterHeight()

`ChartLine visualization.fs2d.TimeLine2D.getYmaxWaterHeight ()`

Returns a chart line showing the water height at the top border during the length of the timeline.

Returns

a chart line.

Definition at line 266 of file TimeLine2D.java.

getYminCumulativeDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getYminCumulativeDischarge ()`

Returns a chart line showing the cumulative discharge at the bottom border during the length of the timeline.

Returns

a chart line.

Definition at line 284 of file TimeLine2D.java.

getYminInstantDischarge()

`ChartLine visualization.fs2d.TimeLine2D.getYminInstantDischarge ()`

Returns a chart line showing the instant discharge at the bottom border during the length of the timeline.

Returns

a chart line.

Definition at line 308 of file TimeLine2D.java.

getYminWaterHeight()

`ChartLine visualization.fs2d.TimeLine2D.getYminWaterHeight ()`

Returns a chart line showing the water height at the bottom border during the length of the timeline.

Returns

a chart line.

Definition at line 325 of file TimeLine2D.java.

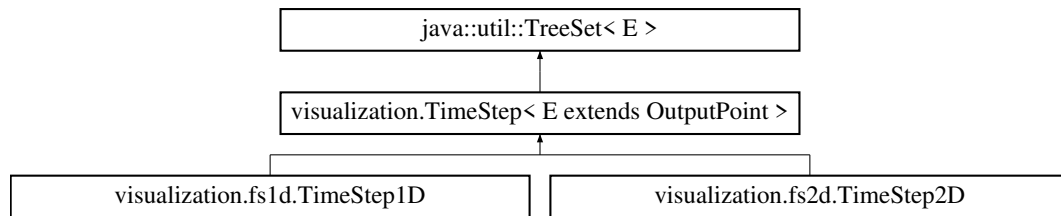
The documentation for this class was generated from the following file:

- `src/visualization/fs2d/TimeLine2D.java`

6.97 visualization.TimeStep< E extends OutputPoint > Class Template Reference

A time step in a FullSWOF output file.

Inheritance diagram for `visualization.TimeStep< E extends OutputPoint >`:



Public Member Functions

- [TimeStep](#) ()
- float [getT](#) ()
- void [setT](#) (float t)

6.97.1 Detailed Description

A time step in a FullSWOF output file.

A time step is a collection of cells.

Parameters

<E>	the type of output points included in the time step
-----	---

Definition at line 72 of file TimeStep.java.

6.97.2 Constructor & Destructor Documentation

TimeStep()

```
visualization.TimeStep< E extends OutputPoint >.TimeStep ( )
```

Constructs an empty time step.

Definition at line 83 of file TimeStep.java.

6.97.3 Member Function Documentation

getT()

```
float visualization.TimeStep< E extends OutputPoint >.getT ( )
```

Returns

the time of the step.

Definition at line 90 of file TimeStep.java.

setT()

```
void visualization.TimeStep< E extends OutputPoint >.setT (
    float t )
```

Sets the time of the step.

Parameters

<i>t</i>	the time of the step
----------	----------------------

Definition at line 99 of file TimeStep.java.

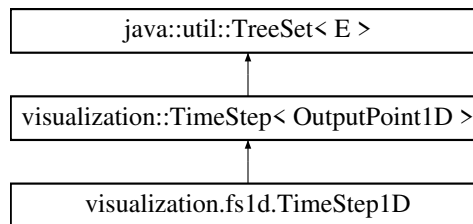
The documentation for this class was generated from the following file:

- [src/visualization/TimeStep.java](#)

6.98 visualization.fs1d.TimeStep1D Class Reference

A time step in a FullSWOF_1D output file.

Inheritance diagram for visualization.fs1d.TimeStep1D:

**Public Member Functions**

- [ChartLine](#) [getChartLineHZ](#) ()
- [ChartContent](#) [getVisualization](#) ()
- [ChartLine](#) [getChartLineZ](#) ()
- [ChartLine](#) [getChartLineU](#) ()
- [ChartLine](#) [getChartLineH](#) ()
- [ChartLine](#) [getChartLineQ](#) ()
- [ChartLine](#) [getChartLineFr](#) ()
- float [getXmaxDischarge](#) ()
- float [getXminDischarge](#) ()
- float [getXminWaterHeight](#) ()
- float [getXmaxWaterHeight](#) ()

6.98.1 Detailed Description

A time step in a FullSWOF_1D output file.

Definition at line 74 of file TimeStep1D.java.

6.98.2 Member Function Documentation**getChartLineFr()**

[ChartLine](#) [visualization.fs1d.TimeStep1D.getChartLineFr](#) ()

Returns the chart line representing the Froude number at all points during this time step.

Returns

a chart line.

Definition at line 196 of file TimeStep1D.java.

getChartLineH()

`ChartLine visualization.fs1d.TimeStep1D.getChartLineH ()`

Returns the chart line representing the free surface at all points during this time step.

Returns

a chart line.

Definition at line 164 of file TimeStep1D.java.

getChartLineHZ()

`ChartLine visualization.fs1d.TimeStep1D.getChartLineHZ ()`

Returns the chart line representing the total height (water and topography) at all points during this time step.

Returns

a chart line.

Definition at line 89 of file TimeStep1D.java.

getChartLineQ()

`ChartLine visualization.fs1d.TimeStep1D.getChartLineQ ()`

Returns the chart line representing the discharge at all points during this time step.

Returns

a chart line.

Definition at line 180 of file TimeStep1D.java.

getChartLineU()

`ChartLine visualization.fs1d.TimeStep1D.getChartLineU ()`

Returns the chart line representing the water velocity at all points during this time step.

Returns

a chart line.

Definition at line 148 of file TimeStep1D.java.

getChartLineZ()

`ChartLine visualization.fs1d.TimeStep1D.getChartLineZ ()`

Returns the chart line representing the topographic height at all points during this time step.

Returns

a chart line.

Definition at line 132 of file TimeStep1D.java.

getVisualization()

```
ChartContent visualization.fs1d.TimeStep1D.getVisualization ( )
```

Return a chart content showing different spatial values of the time step. The values included in the representation are the ones indicated by the boolean variables in the class [visualization.VisualizationPane](#).

Returns

a [ChartContent](#).

Definition at line 106 of file TimeStep1D.java.

getXmaxDischarge()

```
float visualization.fs1d.TimeStep1D.getXmaxDischarge ( )
```

Returns

the discharge at the right boundary during this time step.

Definition at line 208 of file TimeStep1D.java.

getXmaxWaterHeight()

```
float visualization.fs1d.TimeStep1D.getXmaxWaterHeight ( )
```

Returns

the water height at the left boundary during this time step.

Definition at line 229 of file TimeStep1D.java.

getXminDischarge()

```
float visualization.fs1d.TimeStep1D.getXminDischarge ( )
```

Returns

the discharge at the left boundary during this time step.

Definition at line 215 of file TimeStep1D.java.

getXminWaterHeight()

```
float visualization.fs1d.TimeStep1D.getXminWaterHeight ( )
```

Returns

the water height at the right boundary during this time step.

Definition at line 222 of file TimeStep1D.java.

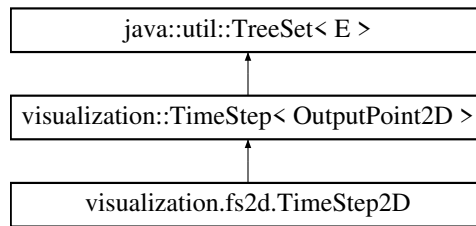
The documentation for this class was generated from the following file:

- [src/visualization/fs1d/TimeStep1D.java](#)

6.99 visualization.fs2d.TimeStep2D Class Reference

A time step in a FullSWOF_2D output file.

Inheritance diagram for visualization.fs2d.TimeStep2D:



Public Member Functions

- SceneGraphComponent [getHSceneGraphComponent](#) ()
- SceneGraphComponent [getUSceneGraphComponent](#) ()
- SceneGraphComponent [getVSceneGraphComponent](#) ()
- SceneGraphComponent [getZSceneGraphComponent](#) ()
- SceneGraphComponent [getHZSceneGraphComponent](#) ()
- SceneGraphComponent [getNSceneGraphComponent](#) ()
- SceneGraphComponent [getFrSceneGraphComponent](#) ()
- SceneGraphComponent [getQxSceneGraphComponent](#) ()
- SceneGraphComponent [getQySceneGraphComponent](#) ()
- SceneGraphComponent [getQSceneGraphComponent](#) ()
- SceneGraphComponent [getVisualization](#) ()
- float [getXmaxDischarge](#) ()
- float [getXminDischarge](#) ()
- float [getYmaxDischarge](#) ()
- float [getYminDischarge](#) ()
- float [getXmaxWaterHeight](#) ()
- float [getXminWaterHeight](#) ()
- float [getYmaxWaterHeight](#) ()
- float [getYminWaterHeight](#) ()

6.99.1 Detailed Description

A time step in a FullSWOF_2D output file.

Definition at line 80 of file TimeStep2D.java.

6.99.2 Member Function Documentation

getFrSceneGraphComponent()

SceneGraphComponent visualization.fs2d.TimeStep2D.getFrSceneGraphComponent ()

Returns a scene graph component showing the Froude number at all points during this time step.

Returns

a scene graph component.

Definition at line 187 of file TimeStep2D.java.

getHSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getHSceneGraphComponent ( )
```

Returns a scene graph component showing the value of H at all points during this time step.

Returns

a scene graph component.

Definition at line 115 of file TimeStep2D.java.

getHZSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getHZSceneGraphComponent ( )
```

Returns a scene graph component showing the value of H+Z at all points during this time step.

Returns

a scene graph component.

Definition at line 163 of file TimeStep2D.java.

getNSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getNSceneGraphComponent ( )
```

Returns a scene graph component showing the value of norm_U at all points during this time step.

Returns

a scene graph component.

Definition at line 175 of file TimeStep2D.java.

getQSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getQSceneGraphComponent ( )
```

Returns a scene graph component showing the value of Q (water discharge Euclidean norm) at all points during this time step.

Returns

a scene graph component.

Definition at line 223 of file TimeStep2D.java.

getQxSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getQxSceneGraphComponent ( )
```

Returns a scene graph component showing the value of Qx at all points during this time step.

Returns

a scene graph component.

Definition at line 199 of file TimeStep2D.java.

getQySceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getQySceneGraphComponent ( )
```

Returns a scene graph component showing the value of Qy at all points during this time step.

Returns

a scene graph component.

Definition at line 211 of file TimeStep2D.java.

getUSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getUSceneGraphComponent ( )
```

Returns a scene graph component showing the value of U at all points during this time step.

Returns

a scene graph component.

Definition at line 127 of file TimeStep2D.java.

getVisualization()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getVisualization ( )
```

Returns a scene graph component showing all points during this time step. The values included in the representation are the ones indicated by the boolean variables in the class [visualization.VisualizationPane](#).

Returns

a scene graph component.

Definition at line 236 of file TimeStep2D.java.

getVSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getVSceneGraphComponent ( )
```

Returns a scene graph component showing the value of V at all points during this time step.

Returns

a scene graph component.

Definition at line 139 of file TimeStep2D.java.

getXmaxDischarge()

```
float visualization.fs2d.TimeStep2D.getXmaxDischarge ( )
```

Returns

the total discharge at the right boundary during this time step.

Definition at line 263 of file TimeStep2D.java.

getXmaxWaterHeight()

```
float visualization.fs2d.TimeStep2D.getXmaxWaterHeight ( )
```

Returns

the average water height at the right boundary during this time step.

Definition at line 524 of file TimeStep2D.java.

getXminDischarge()

```
float visualization.fs2d.TimeStep2D.getXminDischarge ( )
```

Returns

the total discharge at the left boundary during this time step.

Definition at line 276 of file TimeStep2D.java.

getXminWaterHeight()

```
float visualization.fs2d.TimeStep2D.getXminWaterHeight ( )
```

Returns

the average water height at the left boundary during this time step.

Definition at line 541 of file TimeStep2D.java.

getYmaxDischarge()

```
float visualization.fs2d.TimeStep2D.getYmaxDischarge ( )
```

Returns

the total discharge at the top boundary during this time step.

Definition at line 305 of file TimeStep2D.java.

getYmaxWaterHeight()

```
float visualization.fs2d.TimeStep2D.getYmaxWaterHeight ( )
```

Returns

the average water height at the top boundary during this time step.

Definition at line 558 of file TimeStep2D.java.

getYminDischarge()

```
float visualization.fs2d.TimeStep2D.getYminDischarge ( )
```

Returns

the total discharge at the bottom boundary during this time step.

Definition at line 318 of file TimeStep2D.java.

getYminWaterHeight()

```
float visualization.fs2d.TimeStep2D.getYminWaterHeight ( )
```

Returns

the average water height at the bottom boundary during this time step.

Definition at line 575 of file TimeStep2D.java.

getZSceneGraphComponent()

```
SceneGraphComponent visualization.fs2d.TimeStep2D.getZSceneGraphComponent ( )
```

Returns a scene graph component showing the value of Z at all points during this time step.

Returns

a scene graph component.

Definition at line 151 of file TimeStep2D.java.

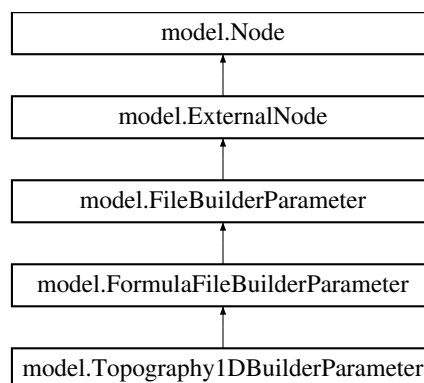
The documentation for this class was generated from the following file:

- src/visualization/fs2d/[TimeStep2D.java](#)

6.100 model.Topography1DBuilderParameter Class Reference

A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z.

Inheritance diagram for model.Topography1DBuilderParameter:

**Public Member Functions**

- [Topography1DBuilderParameter](#) (String [name](#), String [fileName](#), [ExternalNode](#) xLength, [ExternalNode](#) nx-cells)
- String [getFileContent](#) ()

Additional Inherited Members**6.100.1 Detailed Description**

A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z.

Definition at line 70 of file Topography1DBuilderParameter.java.

6.100.2 Constructor & Destructor Documentation

Topography1DBuilderParameter()

```
model.Topography1DBuilderParameter.Topography1DBuilderParameter (
    String name,
    String fileName,
    ExternalNode xLength,
    ExternalNode nxcells )
```

Construct a file builder that writes a topography file for FullSWOF_1D.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file generated
<i>xLength</i>	the node that indicates the value of the xLength parameter
<i>nxcells</i>	the node that indicates the value of the nxcells parameter

Definition at line 95 of file Topography1DBuilderParameter.java.

6.100.3 Member Function Documentation**getFileContent()**

```
String model.Topography1DBuilderParameter.getFileContent ( )
```

Definition at line 105 of file Topography1DBuilderParameter.java.

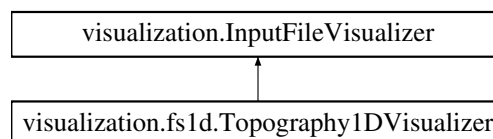
The documentation for this class was generated from the following file:

- [src/model/Topography1DBuilderParameter.java](#)

6.101 visualization.fs1d.Topography1DVisualizer Class Reference

A tool used to get a quick visualization of topography input files for FullSWOF_1D.

Inheritance diagram for visualization.fs1d.Topography1DVisualizer:

**Public Member Functions**

- [Chart getVisualization](#) (File file) throws IOException

6.101.1 Detailed Description

A tool used to get a quick visualization of topography input files for FullSWOF_1D.

The file is represented as a chart with a line topography height.

Definition at line 80 of file Topography1DVisualizer.java.

6.101.2 Member Function Documentation

getVisualization()

`Chart visualization.fs1d.Topography1DVisualizer.getVisualization (File file)` throws `IOException`

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).

Definition at line 82 of file `Topography1DVisualizer.java`.

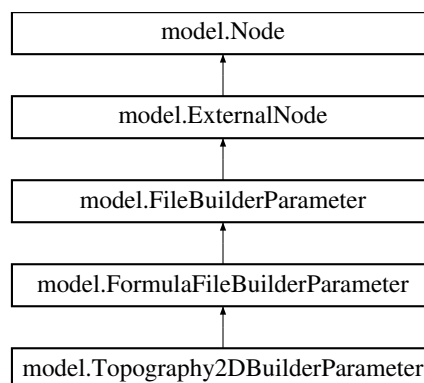
The documentation for this class was generated from the following file:

- `src/visualization/fs1d/Topography1DVisualizer.java`

6.102 model.Topography2DBuilderParameter Class Reference

A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of *z*.

Inheritance diagram for `model.Topography2DBuilderParameter`:

**Public Member Functions**

- [Topography2DBuilderParameter](#) (String *name*, String *fileName*, [ExternalNode](#) *xLength*, [ExternalNode](#) *nx-cells*, [ExternalNode](#) *yLength*, [ExternalNode](#) *nycells*)
- String [getFileContent](#) ()

Additional Inherited Members**6.102.1 Detailed Description**

A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of *z*.

Definition at line 71 of file `Topography2DBuilderParameter.java`.

6.102.2 Constructor & Destructor Documentation

Topography2DBuilderParameter()

```
model.Topography2DBuilderParameter.Topography2DBuilderParameter (
    String name,
    String fileName,
    ExternalNode xLength,
    ExternalNode nxcells,
    ExternalNode yLength,
    ExternalNode nycells )
```

Construct a file builder that writes a topography file for FullSWOF_2D.

Parameters

<i>name</i>	the name of the node
<i>fileName</i>	the name of the file generated
<i>xLength</i>	the node that indicates the value of the xLength parameter
<i>nxcells</i>	the node that indicates the value of the nxcells parameter
<i>yLength</i>	the node that indicates the value of the yLength parameter
<i>nycells</i>	the node that indicates the value of the nycells parameter

Definition at line 110 of file Topography2DBuilderParameter.java.

6.102.3 Member Function Documentation

getFileContent()

```
String model.Topography2DBuilderParameter.getFileContent ( )
```

Definition at line 124 of file Topography2DBuilderParameter.java.

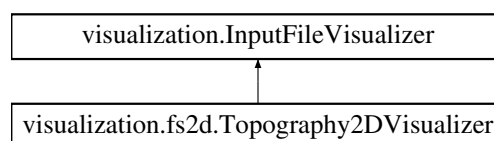
The documentation for this class was generated from the following file:

- src/model/[Topography2DBuilderParameter.java](#)

6.103 visualization.fs2d.Topography2DVisualizer Class Reference

A tool used to get a quick visualization of topography input for FullSWOF_2D.

Inheritance diagram for visualization.fs2d.Topography2DVisualizer:



Public Member Functions

- Component [getVisualization](#) (File file) throws IOException

6.103.1 Detailed Description

A tool used to get a quick visualization of topography input for FullSWOF_2D.

The file is represented as a 3D surface showing topography.

Definition at line 87 of file Topography2DVisualizer.java.

6.103.2 Member Function Documentation

getVisualization()

Component visualization.fs2d.Topography2DVisualizer.getVisualization (
 File *file*) throws IOException

Returns a visualization component to visualize an input file.

Parameters

<i>file</i>	the file to visualize
-------------	-----------------------

Returns

a visualization component.

Exceptions

<i>IOException</i>	if an error occurred while reading the file or if the file format is incorrect
--------------------	--

Implements [visualization.InputFileVisualizer](#).

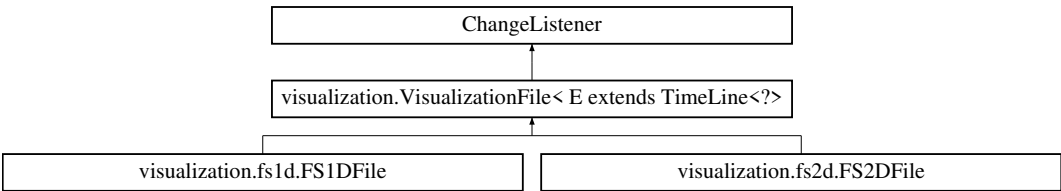
Definition at line 89 of file Topography2DVisualizer.java.

The documentation for this class was generated from the following file:

- src/visualization/fs2d/[Topography2DVisualizer.java](#)

6.104 visualization.VisualizationFile< E extends TimeLine<?> Class Template Reference

Inheritance diagram for visualization.VisualizationFile< E extends TimeLine<?>:



Public Member Functions

- String [getName](#) ()
- E [getTimeline](#) ()
- [VisualizationPane](#) [getView](#) ()
- void [startWatching](#) ()
- void [stateChanged](#) (ChangeEvent e)

- void `stopWatching ()`
- void `updateView ()`

Static Public Member Functions

- static final `VisualizationFile<?> open` (File f) throws IOException

Protected Member Functions

- abstract void `setUpView ()`

Protected Attributes

- String `name`
The name of the file.
- `OutputFileReader` `reader`
The reader used to read the file.
- `E` `timeline`
The time line created with the file.
- `VisualizationPane` `view`
The visualization pane used to display this file.

6.104.1 Detailed Description

- A FullSWOF output file

Parameters

<code><E></code>	the type of time line associated with this file
------------------------	---

Definition at line 91 of file VisualizationFile.java.

6.104.2 Member Function Documentation

`getName()`

String `visualization.VisualizationFile< E extends TimeLine<?>.getName ()`

Returns

the name of the file.

Definition at line 296 of file VisualizationFile.java.

`getTimeline()`

`E visualization.VisualizationFile< E extends TimeLine<?>.getTimeline ()`

Returns

the time line created with the file.

Definition at line 303 of file VisualizationFile.java.

getView()

```
VisualizationPane visualization.VisualizationFile< E extends TimeLine<?>.getView ( )
```

Returns

the visualization pane used to display this file.

Definition at line 310 of file VisualizationFile.java.

open()

```
static final VisualizationFile<?> visualization.VisualizationFile< E extends TimeLine<?>.open  
(  
    File f ) throws IOException [static]
```

Opens a FullSWOF output file, trying to determine its format. Format supported are Gnuplot files for FullS↔
WOF_1D and FullSWOF_2D, and VTK files for FullSWOF_2D.

Parameters

<i>f</i>	the file to open
----------	------------------

Returns

a VisualisationFile instance of the file.

Exceptions

<i>IOException</i>	if an error occurs while reading the file, or if the format could not be determined
--------------------	---

Definition at line 107 of file VisualizationFile.java.

setUpView()

```
abstract void visualization.VisualizationFile< E extends TimeLine<?>.setUpView ( ) [abstract],  
[protected]
```

Sets up the visualization pane used to display this file.

startWatching()

```
void visualization.VisualizationFile< E extends TimeLine<?>.startWatching ( )
```

Watch changes on the physical file and and modify the time line accordingly.

Definition at line 319 of file VisualizationFile.java.

stateChanged()

```
void visualization.VisualizationFile< E extends TimeLine<?>.stateChanged (   
    ChangeEvent e )
```

Called when the timeline is modified after modifications on the physical file.

Parameters

e	the triggering event
---	----------------------

Definition at line 332 of file VisualizationFile.java.

stopWatching()

```
void visualization.VisualizationFile< E extends TimeLine<?>.stopWatching ( )
```

Stops watching the physical file.

Definition at line 340 of file VisualizationFile.java.

updateView()

```
void visualization.VisualizationFile< E extends TimeLine<?>.updateView ( )
```

Updates the file view or builds it if necessary. Must be called after the timeline has been updated.

Definition at line 349 of file VisualizationFile.java.

6.104.3 Member Data Documentation**name**

```
String visualization.VisualizationFile< E extends TimeLine<?>.name [protected]
```

The name of the file.

Displayed in the visualization frame title.

Definition at line 276 of file VisualizationFile.java.

reader

```
OutputFileReader visualization.VisualizationFile< E extends TimeLine<?>.reader [protected]
```

The reader used to read the file.

Definition at line 281 of file VisualizationFile.java.

timeline

```
E visualization.VisualizationFile< E extends TimeLine<?>.timeline [protected]
```

The time line created with the file.

Definition at line 286 of file VisualizationFile.java.

view

```
VisualizationPane visualization.VisualizationFile< E extends TimeLine<?>.view [protected]
```

The visualization pane used to display this file.

Definition at line 291 of file VisualizationFile.java.

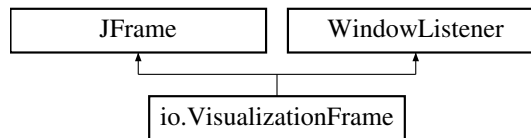
The documentation for this class was generated from the following file:

- src/visualization/VisualizationFile.java

6.105 io.VisualizationFrame Class Reference

An instance of this class is a JFrame used to visualize a FullSWOF output file.

Inheritance diagram for io.VisualizationFrame:



Public Member Functions

- [VisualizationFrame](#) ([VisualizationFile](#)<?> file)
- void [windowActivated](#) (WindowEvent evt)
- void [windowClosed](#) (WindowEvent evt)
- void [windowClosing](#) (WindowEvent evt)
- void [windowDeactivated](#) (WindowEvent evt)
- void [windowDeiconified](#) (WindowEvent evt)
- void [windowIconified](#) (WindowEvent evt)
- void [windowOpened](#) (WindowEvent evt)

6.105.1 Detailed Description

An instance of this class is a JFrame used to visualize a FullSWOF output file.

Definition at line 73 of file VisualizationFrame.java.

6.105.2 Constructor & Destructor Documentation

VisualizationFrame()

```
io.VisualizationFrame.VisualizationFrame (
    VisualizationFile<?> file )
```

Constructs and displays a visualization window for a FullSWOF output file

Parameters

<i>file</i>	the file to visualize.
-------------	------------------------

Definition at line 91 of file VisualizationFrame.java.

6.105.3 Member Function Documentation

windowActivated()

```
void io.VisualizationFrame.windowActivated (
    WindowEvent evt )
```

Called when the main window is activated. No effect.

Definition at line 112 of file VisualizationFrame.java.

windowClosed()

```
void io.VisualizationFrame.windowClosed (
    WindowEvent evt )
```

Called when the main window is closed. No effect.

Definition at line 119 of file VisualizationFrame.java.

windowClosing()

```
void io.VisualizationFrame.windowClosing (
    WindowEvent evt )
```

Called when the user attempts to close the window in any way. Displays a confirmation message before closing the window.

Definition at line 127 of file VisualizationFrame.java.

windowDeactivated()

```
void io.VisualizationFrame.windowDeactivated (
    WindowEvent evt )
```

Called when the main window is deactivated. No effect.

Definition at line 140 of file VisualizationFrame.java.

windowDeiconified()

```
void io.VisualizationFrame.windowDeiconified (
    WindowEvent evt )
```

Called when the main window is deiconified. No effect.

Definition at line 147 of file VisualizationFrame.java.

windowIconified()

```
void io.VisualizationFrame.windowIconified (
    WindowEvent evt )
```

Called when the main window is iconified. No effect.

Definition at line 154 of file VisualizationFrame.java.

windowOpened()

```
void io.VisualizationFrame.windowOpened (
    WindowEvent evt )
```

Called when the main window is opened. No effect.

Definition at line 161 of file VisualizationFrame.java.

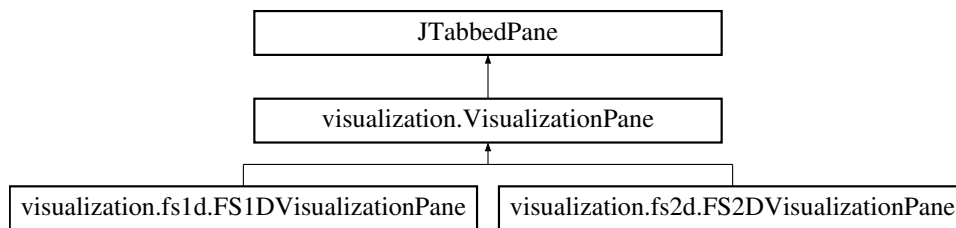
The documentation for this class was generated from the following file:

- [src/io/VisualizationFrame.java](#)

6.106 visualization.VisualizationPane Class Reference

A tabbed pane presenting a [VisualizationFile](#).

Inheritance diagram for visualization.VisualizationPane:



Public Member Functions

- abstract void [update](#) ()

Static Public Attributes

- static boolean [showH](#)
Indicates whether H values should be included in spatial visualization.
- static boolean [showU](#)
Indicates whether U values should be included in spatial visualization.
- static boolean [showV](#)
Indicates whether V values should be included in spatial visualization.
- static boolean [showZ](#)
Indicates whether Z values should be included in spatial visualization.
- static boolean [showHZ](#)
Indicates whether H+Z values should be included in spatial visualization.
- static boolean [showN](#)
Indicates whether norm_U values should be included in spatial visualization.
- static boolean [showFr](#)
Indicates whether Fr values should be included in spatial visualization.
- static boolean [showQx](#)
Indicates whether Qx values should be included in spatial visualization.
- static boolean [showQy](#)
Indicates whether Qy values should be included in spatial visualization.
- static boolean [showQ](#)
Indicates whether Q values should be included in spatial visualization.
- static Color [hColor](#)
The color used to display H values in spatial visualization.
- static Color [uColor](#)
The color used to display U values in spatial visualization.
- static Color [vColor](#)
The color used to display V values in spatial visualization.
- static Color [zColor](#)
The color used to display Z values in spatial visualization.
- static Color [hzColor](#)
The color used to display H+Z values in spatial visualization.

- static Color `nColor`
The color used to display norm_U values in spatial visualization.
- static Color `frColor`
The color used to display Fr values in spatial visualization.
- static Color `qxColor`
The color used to display Qx values in spatial visualization.
- static Color `qyColor`
The color used to display Qy values in spatial visualization.
- static Color `qColor`
The color used to display Q values in spatial visualization.
- static Color `defaultLineColor`
The standard color used for chart lines.

6.106.1 Detailed Description

A tabbed pane presenting a [VisualizationFile](#).

Definition at line 69 of file VisualizationPane.java.

6.106.2 Member Function Documentation

`update()`

```
abstract void visualization.VisualizationPane.update ( ) [abstract]
```

Update the content of the tabbed pane to reflect changes on the file.

6.106.3 Member Data Documentation

`defaultLineColor`

```
Color visualization.VisualizationPane.defaultLineColor [static]
```

The standard color used for chart lines.

Definition at line 174 of file VisualizationPane.java.

`frColor`

```
Color visualization.VisualizationPane.frColor [static]
```

The color used to display Fr values in spatial visualization.

Definition at line 154 of file VisualizationPane.java.

`hColor`

```
Color visualization.VisualizationPane.hColor [static]
```

The color used to display H values in spatial visualization.

Definition at line 124 of file VisualizationPane.java.

hzColor

`Color visualization.VisualizationPane.hzColor [static]`
The color used to display H+Z values in spatial visualization.
Definition at line 144 of file VisualizationPane.java.

nColor

`Color visualization.VisualizationPane.nColor [static]`
The color used to display norm_U values in spatial visualization.
Definition at line 149 of file VisualizationPane.java.

qColor

`Color visualization.VisualizationPane.qColor [static]`
The color used to display Q values in spatial visualization.
Definition at line 169 of file VisualizationPane.java.

qxColor

`Color visualization.VisualizationPane.qxCOLOR [static]`
The color used to display Qx values in spatial visualization.
Definition at line 159 of file VisualizationPane.java.

qyColor

`Color visualization.VisualizationPane.qyColor [static]`
The color used to display Qy values in spatial visualization.
Definition at line 164 of file VisualizationPane.java.

showFr

`boolean visualization.VisualizationPane.showFr [static]`
Indicates whether Fr values should be included in spatial visualization.
Definition at line 104 of file VisualizationPane.java.

showH

`boolean visualization.VisualizationPane.showH [static]`
Indicates whether H values should be included in spatial visualization.
Definition at line 74 of file VisualizationPane.java.

showHZ

`boolean visualization.VisualizationPane.showHZ [static]`
Indicates whether H+Z values should be included in spatial visualization.
Definition at line 94 of file VisualizationPane.java.

showN

```
boolean visualization.VisualizationPane.showN [static]
```

Indicates whether norm_U values should be included in spatial visualization.
Definition at line 99 of file VisualizationPane.java.

showQ

```
boolean visualization.VisualizationPane.showQ [static]
```

Indicates whether Q values should be included in spatial visualization.
Definition at line 119 of file VisualizationPane.java.

showQx

```
boolean visualization.VisualizationPane.showQx [static]
```

Indicates whether Qx values should be included in spatial visualization.
Definition at line 109 of file VisualizationPane.java.

showQy

```
boolean visualization.VisualizationPane.showQy [static]
```

Indicates whether Qy values should be included in spatial visualization.
Definition at line 114 of file VisualizationPane.java.

showU

```
boolean visualization.VisualizationPane.showU [static]
```

Indicates whether U values should be included in spatial visualization.
Definition at line 79 of file VisualizationPane.java.

showV

```
boolean visualization.VisualizationPane.showV [static]
```

Indicates whether V values should be included in spatial visualization.
Definition at line 84 of file VisualizationPane.java.

showZ

```
boolean visualization.VisualizationPane.showZ [static]
```

Indicates whether Z values should be included in spatial visualization.
Definition at line 89 of file VisualizationPane.java.

uColor

```
Color visualization.VisualizationPane.uColor [static]
```

The color used to display U values in spatial visualization.
Definition at line 129 of file VisualizationPane.java.

vColor

Color visualization.VisualizationPane.vColor [static]

The color used to display V values in spatial visualization.

Definition at line 134 of file VisualizationPane.java.

zColor

Color visualization.VisualizationPane.zColor [static]

The color used to display Z values in spatial visualization.

Definition at line 139 of file VisualizationPane.java.

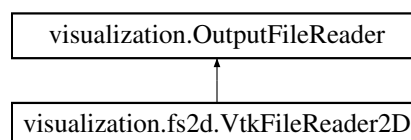
The documentation for this class was generated from the following file:

- src/visualization/[VisualizationPane.java](#)

6.107 visualization.fs2d.VtkFileReader2D Class Reference

A reader for VTK output files generated by FullSWOF_2D.

Inheritance diagram for visualization.fs2d.VtkFileReader2D:

**Public Member Functions**

- [VtkFileReader2D](#) (File file, [TimeLine2D timeline](#)) throws IOException
- void [startWatching](#) ()
- void [stopWatching](#) ()
- void [updateTimeline](#) () throws IOException

Additional Inherited Members**6.107.1 Detailed Description**

A reader for VTK output files generated by FullSWOF_2D.

Definition at line 72 of file VtkFileReader2D.java.

6.107.2 Constructor & Destructor Documentation**VtkFileReader2D()**

```
visualization.fs2d.VtkFileReader2D.VtkFileReader2D (
    File file,
    TimeLine2D timeline ) throws IOException
```

Builds a reader for VTK output files generated by FullSWOF_2D. Unlike Gnuplot files, VTK takes multiple files to save a timeline. Each file is a time step. Opening a file with this method will actually open all the files in the same directory that are named huz_evolution.datx.vtk, where x is the number of the time step.

Parameters

<i>file</i>	the file or directory to read
<i>timeline</i>	the timeline to update

Exceptions

<i>IOException</i>	if an error occurs while reading the file
--------------------	---

Definition at line 99 of file VtkFileReader2D.java.

6.107.3 Member Function Documentation**startWatching()**

```
void visualization.fs2d.VtkFileReader2D.startWatching ( )
```

Start watching the physical for changes and update the timeline accordingly.
Definition at line 115 of file VtkFileReader2D.java.

stopWatching()

```
void visualization.fs2d.VtkFileReader2D.stopWatching ( )
```

Stop watching the physical file.
Definition at line 124 of file VtkFileReader2D.java.

updateTimeline()

```
void visualization.fs2d.VtkFileReader2D.updateTimeline ( ) throws IOException
```

Modify the timeline so that its content reflects that of the physical file.

Exceptions

<i>IOException</i>	if an error occurs while reading
--------------------	----------------------------------

Definition at line 144 of file VtkFileReader2D.java.

The documentation for this class was generated from the following file:

- src/visualization/fs2d/[VtkFileReader2D.java](#)

Chapter 7

File Documentation

7.1 src/io/DataSetBuilderDialog.java File Reference

Dialog box.

Classes

- class [io.DataSetBuilderDialog](#)

A dialog box that allows the user to add custom data to a chart from a file.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.1.1 Detailed Description

Dialog box.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A dialog box that allows the user to add custom data to a chart from a file. The dialog contains fields to indicate the path of the file, the display name of the data, its type of display (line or points, color).

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.2 src/io/FullswofIO.java File Reference

Handle interactions with FullSWOF.

Classes

- class [io.FullswofIO](#)

This class provides static methods to handle the interactions with the C++ FullSWOF code.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.2.1 Detailed Description

Handle interactions with FullSWOF.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This class provides static methods to handle the interactions with the C++ FullSWOF code.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.3 src/io/HtmlAbout.java File Reference

Displays the content of About.

Classes

- class [io.HtmlAbout](#)

An instance of this class is a JFrame used to display the content of About.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.3.1 Detailed Description

Displays the content of About.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

An instance of this class is a JFrame used to display the content of About.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.4 src/io/HtmlFrame.java File Reference

How to display the content of an html file.

Classes

- class [io.HtmlFrame](#)

An instance of this class is a JFrame used to display the content of any HTML with basic style support.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.4.1 Detailed Description

How to display the content of an html file.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An instance of this class is a JFrame used to display the content of any HTML with basic style support. It is useful to display the user manual or the application credits.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.5 src/io/MainFrame.java File Reference

Main window.

Classes

- class [io.MainFrame](#)

An instance of this class is a JFrame corresponding to the main window of the user interface.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.5.1 Detailed Description

Main window.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An instance of this class is a JFrame corresponding to the main window of the user interface. It includes the main menu and the area used to set the FullSWOF parameters.

See also

`javax.swing.JFrame`

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.6 src/io/package-info.java File Reference

Package for the static parts of the interface.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.6.1 Detailed Description

Package for the static parts of the interface.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides the classes needed to build the static parts of the user interface. This includes classes to build windows and their menus using swing components, but also the procedures to open and save files, run FullSWOF and render its output.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.7 src/model/definition/package-info.java File Reference

Package to define FullSWOF configurations.

Packages

- package [model.definition](#)

Each class in this package provides a single static method to instantiate a model tree corresponding to a FullSWOF configuration.

7.7.1 Detailed Description

Package to define FullSWOF configurations.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

Each class in this package provides a single static method to instantiate a model tree corresponding to a FullSWOF configuration. To create a new configuration, create a new class with a static method to instantiate a node. Take for example the classes already provided in this package. To make the new configuration available in the user interface, you only need to add it to the array `availableConfigurations` in [io.Procedures](#).

FullSWOF_UI is entirely internationalized. Each configuration should follow this pattern and you should avoid hardcoding the parameters name and description in the class. Instead use a `resourceBundle`, for which you must provide at least one default file. Place this file in a directory used only for this configuration, preferably in the `I10n/config` directory. You can provide more than one localization for a configuration, but new languages will only be displayed if the languages are also available for the user interface. For example if the user interface is localized in English and French, and you provide English and German localizations for a configuration, only the English localization will ever be used.

See also

`java.util.ResourceBundle`
[model](#)

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.8 src/model/package-info.java File Reference

Package for the parameters.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.8.1 Detailed Description

Package for the parameters.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides the necessary classes to build a model for FullSWOF_UI parameters. A model is organized as a composite tree, where all the objects extend the abstract class Node. There are two types of nodes : internal nodes can have child nodes while external nodes cannot. External nodes usually represent parameters used in the interface, while internal nodes are groups of parameters (or groups of groups, since there is not limit to the depth of the tree). This package offers several implementations of external nodes, such as numeric parameters, string parameters, multiple choice parameters and many others, but you may need to implement new classes for more unusual purposes.

Besides nodes, this package also provides a [model.Dependency](#) abstract class. A dependency is a binary relationship between two external nodes that allow an action to be triggered under certain conditions.

See also

[model.definition](#) to see examples of model trees
[ui](#) for the corresponding controllers and views

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.9 src/parser/package-info.java File Reference

Package for parsing the mathematical formulas.

Packages

- package [parser](#)

This package contains the parser and lexer used to parse mathematical formulas.

7.9.1 Detailed Description

Package for parsing the mathematical formulas.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package contains the parser and lexer used to parse mathematical formulas. These classes were auto-generated by ANTLR v3.4. Some methods inherited from BaseRecognizer are overridden so that exceptions are thrown instead of the default recovery mechanism.

See also

the file `Formula.g` for the grammar definition

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.10 `src/ui/package-info.java` File Reference

Package for the controllers.

Packages

- package `ui`

This package provides the controllers and views associated with the model classes.

7.10.1 Detailed Description

Package for the controllers.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides the controllers and views associated with the model classes. Controllers and views are closely linked, so there are no classes for views. Instead, each Controller has a view attribute, which it has to instantiate.

The controllers maintain a hierarchy which is parallel to that of the model. Internal node controllers must therefore maintain a list of child controllers, similar to the list of child nodes of their node. Likewise, views are modeled as a tree of JComponents.

The view never updates the model itself, it must fire an event (usually a `FocusEvent` to indicate that the view has lost focus after the user has finished writing his input) to the controller that will update the model. On the other hand, the model must fire a `ChangeEvent` to notify the controller that will update the view. Note that these procedures only apply to external nodes and their controllers, since internal nodes are not modified after their instantiation.

See also

`javax.swing.JComponent`

`javax.swing.event`

`model`

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.11 src/visualization/fs1d/package-info.java File Reference

Package for FullSWOF_1D visualization.

Packages

- package [visualization.fs1d](#)

This package provides the classes used to visualize FullSWOF_1D output files.

7.11.1 Detailed Description

Package for FullSWOF_1D visualization.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides the classes used to visualize FullSWOF_1D output files. The only format currently supported is the Gnuplot file format.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.12 src/visualization/fs2d/package-info.java File Reference

Package for FullSWOF_2D visualization.

Packages

- package [visualization.fs2d](#)

This package provides the classes used to visualize FullSWOF_2D output files.

7.12.1 Detailed Description

Package for FullSWOF_2D visualization.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides the classes used to visualize FullSWOF_2D output files. Gnuplot files and VTK files are supported.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.13 src/visualization/package-info.java File Reference

Package for the visualization.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.13.1 Detailed Description

Package for the visualization.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This package provides classes used to visualize FullSWOF output files.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.14 src/io/PreferencesDialog.java File Reference

Preferences window.

Classes

- class [io.PreferencesDialog](#)

An instance of this class is a JDialog corresponding to the preferences window of the user interface.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.14.1 Detailed Description

Preferences window.

Author

Simon Robillard (2012)
Frédéric Darboux (2013)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

An instance of this class is a JDialog corresponding to the preferences window of the user interface. The preferences are saved in a file named settings.properties, which is located in the hidden directory `./fullswof_ui` in the user directory.

See also

`javax.swing.JDialog`

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.15 src/io/Procedures.java File Reference

Static methods.

Classes

- class [io.Procedures](#)

This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.15.1 Detailed Description

Static methods.

Author

Simon Robillard (2012)
Antoine Schellenberger (2015)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

This class provides static methods used by the user interface, most notably for opening and saving files, or creating a new project.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.16 src/io/ProgressDialog.java File Reference

Dialog box with progress bar.

Classes

- class [io.ProgressDialog](#)
A dialog box including a progress bar, a console display and a cancel button.

Packages

- package [io](#)
This package provides the classes needed to build the static parts of the user interface.

7.16.1 Detailed Description

Dialog box with progress bar.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A dialog box including a progress bar, a console display and a cancel button. This dialog box is linked to a thread that can be interrupted by pressing the button.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.17 src/io/Start.java File Reference

Start.

Classes

- class [io.Start](#)

The executable class used to launch the application.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.17.1 Detailed Description

Start.

Author

Simon Robillard (2012)

Antoine Schellenberger (2015)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

The executable class used to launch the application.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.18 src/io/VisualizationFrame.java File Reference

Visualization.

Classes

- class [io.VisualizationFrame](#)

An instance of this class is a JFrame used to visualize a FullSWOF output file.

Packages

- package [io](#)

This package provides the classes needed to build the static parts of the user interface.

7.18.1 Detailed Description

Visualization.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An instance of this class is a JFrame used to visualize a FullSWOF output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.19 src/model/BoundaryFileParameter.java File Reference

Builds a boundary file.

Classes

- class [model.BoundaryFileParameter](#)

A parameter used to build a boundary file.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.19.1 Detailed Description

Builds a boundary file.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A parameter used to build a boundary file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.20 src/model/BoundaryParameter.java File Reference

Stores a boundary file pathname.

Classes

- class [model.BoundaryParameter](#)
A parameter used to store a boundary file pathname.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.20.1 Detailed Description

Stores a boundary file pathname.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A parameter used to store a boundary file pathname.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.21 src/model/definition/Definition_1D.java File Reference

FullSWOF_1D configuration.

Classes

- class [model.definition.Definition_1D](#)

This class provides a static method for generating the configuration used by FullSWOF_1D parameters files.

Packages

- package [model.definition](#)

Each class in this package provides a single static method to instantiate a model tree corresponding to a FullSWOF configuration.

7.21.1 Detailed Description

FullSWOF_1D configuration.

Author

Simon Robillard (2012)

Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

This class provides a static method for generating the configuration used by FullSWOF_1D parameters files.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.22 src/model/definition/Definition_2D.java File Reference

FullSWOF_2D configuration.

Classes

- class [model.definition.Definition_2D](#)

This class provides a static method for generating the configuration used by FullSWOF_2D parameters files.

Packages

- package [model.definition](#)

Each class in this package provides a single static method to instantiate a model tree corresponding to a FullSWOF configuration.

7.22.1 Detailed Description

FullSWOF_2D configuration.

Author

Simon Robillard (2012)

Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

This class provides a static method for generating the configuration used by FullSWOF_2D parameters files.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.23 src/model/Dependency.java File Reference

Dependencies implementation.

Classes

- class [model.Dependency](#)

A dependency is a binary relationship between external nodes.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.23.1 Detailed Description

Dependencies implementation.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A dependency is a binary relationship between external nodes. Every time the value of the master node is changed, it is compared to the target value of the dependency. If the new value of the master node is equal to that target value, a reaction is triggered on the slave node, which depends on the particular implementation of the dependency. Dependencies are very useful to use with a multiple choice parameter as the master node, where a certain choice will trigger changes on other parameters, such as disabling or enabling them.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.24 src/model/DirectoryExtensionParameter.java File Reference

Directory extension name used in FullSWOF.

Classes

- class [model.DirectoryExtensionParameter](#)

A directory extension parameter is a special implementation of [ExternalNode](#), which is typically used only once in a configuration tree.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.24.1 Detailed Description

Directory extension name used in FullSWOF.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A directory extension parameter is a special implementation of `ExternalNode`, which is typically used only once in a configuration tree. It is used to indicate the suffix of the outputs folder used by FullSWOF, which will be named `"Outputs"+this.value`.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.25 src/model/DisablingDependency.java File Reference

Disables dependencies.

Classes

- class [model.DisablingDependency](#)

A disabling dependency is used to disable the slave node when the master node is set to the target value.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.25.1 Detailed Description

Disables dependencies.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A disabling dependency is used to disable the slave node when the master node is set to the target value. Note that the slave node will not be automatically enabled if the master node value changes later. An enabling dependency should therefore be added on the master node for each other possible value. Without this, this user will have no mean to re-enable the parameters which was disabled.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.26 src/model/EnablingDependency.java File Reference

Enables dependencies.

Classes

- class [model.EnablingDependency](#)

An enabling dependency is used to enable the slave node when the master node is set to the target value.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.26.1 Detailed Description

Enables dependencies.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

An enabling dependency is used to enable the slave node when the master node is set to the target value. All nodes are created enabled by default, so you need to create an enabling dependency only if a disabling dependency has been added to the same slave node.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.27 src/model/ExternalNode.java File Reference

External node.

Classes

- class [model.ExternalNode](#)

An external node in the tree model, typically a FullSWOF parameter.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.27.1 Detailed Description

External node.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

An external node in the tree model, typically a FullSWOF parameter. An external node does not have any child nodes.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.28 src/model/FieldParameter.java File Reference

Implements FullSWOF parameters.

Classes

- class [model.FieldParameter](#)

This class provides the most permissive implementation of an external node, as any value will be considered a valid entry.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.28.1 Detailed Description

Implements FullSWOF parameters.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This class provides the most permissive implementation of an external node, as any value will be considered a valid entry.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.29 src/model/FileBuilderParameter.java File Reference

Implements parameters that are not in FullSWOF.

Classes

- class [model.FileBuilderParameter](#)
A parameter used to create an annex file.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.29.1 Detailed Description

Implements parameters that are not in FullSWOF.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A parameter used to create an annex file. Unlike other external nodes, a file builder parameter does not appear in the parameters.txt file. They are used to write an annex file such as a rain, topography or huv file, upon validation by the controller. Note that this external node does not have a value, so operations dealing with the node value (including dependencies) will throw an exception.

See also

[ui.FileBuilderController](#)

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.30 src/model/FileParameter.java File Reference

Parameter that stores a file pathname.

Classes

- class [model.FileParameter](#)
A parameter used to store a file pathname.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.30.1 Detailed Description

Parameter that stores a file pathname.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A parameter used to store a file pathname. The file must be declared by its absolute pathname and must exist at that location. You can optionally specify a visualization tool for this parameter. In this case, the view will provide a way for the user to get a quick visualization of the file content (for example a chart).

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.31 src/model/FloatParameter.java File Reference

Floating point parameter.

Classes

- class [model.FloatParameter](#)
A parameter with a floating point number value.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.31.1 Detailed Description

Floating point parameter.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A parameter with a floating point number value. The value is still stored as a string, which is parsed to check whether it represents a numeric value. The acceptance interval of the value can be specified in the constructor or omitted, in which case any real number will be considered valid.

See also

`java.lang.Float.valueOf(String s)` to learn more about the lexical syntax rules for writing a floating point number as a string.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.32 src/model/FormulaFileBuilderParameter.java File Reference

Parameter with mathematical formulas.

Classes

- class [model.FormulaFileBuilderParameter](#)

A file builder parameter that uses a set of mathematical formulas to build a file.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.32.1 Detailed Description

Parameter with mathematical formulas.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A file builder parameter that uses a set of mathematical formulas to build a file. The actual use of these formulas depend on the implementation

See also

[parser](#)

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.33 src/model/HU1DBuilderParameter.java File Reference

Writes hu files for FullSWOF_1D.

Classes

- class [model.HU1DBuilderParameter](#)

A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.33.1 Detailed Description

Writes hu files for FullSWOF_1D.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A file builder that writes a HU file for FullSWOF_1D, using parsed formulas to determine the value of h and u.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.34 src/model/HUV2DBuilderParameter.java File Reference

Writes huv files for FullSWOF_2D.

Classes

- class [model.HUV2DBuilderParameter](#)

A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.34.1 Detailed Description

Writes huv files for FullSWOF_2D.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A file builder that writes a HUV file for FullSWOF_2D, using parsed formulas to determine the value of h, u and v.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.35 src/model/IntegerParameter.java File Reference

Integer parameter.

Classes

- class [model.IntegerParameter](#)
A parameter with an integer value.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.35.1 Detailed Description

Integer parameter.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A parameter with an integer value. The value is still stored as a string, which is parsed to check if whether it represents a numeric value. The acceptance interval of the value can be specified in the constructor or omitted, in which case any integer will be considered valid.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.36 src/model/InternalNode.java File Reference

Internal node.

Classes

- class [model.InternalNode](#)

This class can be used for any internal node of the tree.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.36.1 Detailed Description

Internal node.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

This class can be used for any internal node of the tree. The standard controller instantiate the view as a panel.

See also

[ui.ParametersGroupController](#)

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.37 src/model/Interval.java File Reference

Numerical interval.

Classes

- class [model.Interval](#)

Describes a numerical interval.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.37.1 Detailed Description

Numerical interval.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

Describes a numerical interval. The interval is closed by default but the inclusion of each endpoint can be specified in the constructor

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.38 src/model/ListFile.java File Reference

For tables.

Classes

- class [model.ListFile](#)

The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : [RainFileParameter](#), [PointFileParametre](#) , [BoundaryFileparameter](#).

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.38.1 Detailed Description

For tables.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

The table is a parameter used to create an annex file All tables where It need checked that the table is valid Like : RainFileParameter, PointFileParametre , BoundaryFileparameter.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.39 src/model/MultipleChoiceParameter.java File Reference

Choice parameter (list of possible values)

Classes

- class [model.MultipleChoiceParameter](#)
A parameter with a finite set of accepted values.
- class [model.MultipleChoiceParameter.PossibleValue](#)
A possible value is constituted of two strings.

Packages

- package [model](#)
This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.39.1 Detailed Description

Choice parameter (list of possible values)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A parameter with a finite set of accepted values.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.40 src/model/Node.java File Reference

Node.

Classes

- class [model.Node](#)

A node in the model tree.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.40.1 Detailed Description

Node.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A node in the model tree. According to the composite design pattern of the model, this abstract class is extended by almost all classes from the package model, except for the [model.Dependency](#) class and classes that extend it.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.41 src/model/PointFileParameter.java File Reference

Builds a point file.

Classes

- class [model.PointFileParameter](#)

A parameter used to build a point file.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.41.1 Detailed Description

Builds a point file.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A parameter used to build a point file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.42 src/model/RainFileParameter.java File Reference

Parameter for rain file.

Classes

- class [model.RainFileParameter](#)

A parameter used to build a rain file.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.42.1 Detailed Description

Parameter for rain file.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A parameter used to build a rain file.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.43 src/model/RootNode.java File Reference

Node (preferentially root node)

Classes

- class [model.RootNode](#)

This class can be used for any internal node, but its controller is better suited to the root of the tree.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.43.1 Detailed Description

Node (preferentially root node)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This class can be used for any internal node, but its controller is better suited to the root of the tree. The root controller offers a tabbed pane view, where each child node is a tab.

See also

[ui.RootController](#)

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.44 src/model/SettingDependency.java File Reference

Sets the value of the slave node.

Classes

- class [model.SettingDependency](#)

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.44.1 Detailed Description

Sets the value of the slave node.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.45 src/model/SettingDependency2.java File Reference

Sets the value of the slave node taking into account error values.

Classes

- class [model.SettingDependency2](#)

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.45.1 Detailed Description

Sets the value of the slave node taking into account error values.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A setting dependency is used to set the slave node to a particular value when the master node is set to the target value.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.46 src/model/Topography1DBuilderParameter.java File Reference

Writes a FullSWOF_1D topography file from a formula.

Classes

- class [model.Topography1DBuilderParameter](#)

A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.46.1 Detailed Description

Writes a FullSWOF_1D topography file from a formula.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A file builder that writes a topography file for FullSWOF_1D, using a parsed formula to determine the value of z.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.47 src/model/Topography2DBuilderParameter.java File Reference

Writes a FullSWOF_2D topography file from a formula.

Classes

- class [model.Topography2DBuilderParameter](#)

A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of z.

Packages

- package [model](#)

This package provides the necessary classes to build a model for FullSWOF_UI parameters.

7.47.1 Detailed Description

Writes a FullSWOF_2D topography file from a formula.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A file builder that writes a topography file for FullSWOF_2D, using a parsed formula to determine the value of z.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.48 src/parser/FormulaLexer.java File Reference

Lexer grammar.

Classes

- class [parser.FormulaLexer](#)
Transforms the character stream into a series of tokens.
- class [parser.FormulaLexer.DFA8](#)
- class [parser.FormulaLexer.DFA12](#)

Packages

- package [parser](#)
This package contains the parser and lexer used to parse mathematical formulas.

7.48.1 Detailed Description

Lexer grammar.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-16

Transforms the character stream into a series of tokens. This file has been adapted from the file generated by ANTLR 3.4 from the file Formula.g 2012-06-12 15:43:12.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.49 src/parser/FormulaParser.java File Reference

Parser.

Classes

- class [parser.FormulaParser](#)
Walks though the tokens to form mathematical sentences in the grammar.

Packages

- package [parser](#)
This package contains the parser and lexer used to parse mathematical formulas.

7.49.1 Detailed Description

Parser.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-16

Walks through the tokens to form mathematical sentences in the grammar. This file has been adapted from the file generated by ANTLR 3.4 from the file Formula.g 2012-06-12 15:43:11.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.50 src/ui/BoundaryFileController.java File Reference

Controller for a boundary file builder node.

Classes

- class [ui.BoundaryFileController](#)
A controller for a boundary file builder node.
- class [ui.BoundaryFileController.CellRenderer](#)
An instance of this class is used to render the cells of the table in the view.
- class [ui.BoundaryFileController.BoundaryModel](#)
The model used by the view table.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.50.1 Detailed Description

Controller for a boundary file builder node.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a boundary file builder node. This controller can set up a view that includes an editable table where the user can write time and file (or by brown file) value.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.51 src/ui/BoundaryFileParameterController.java File Reference

Controller for a boundary file parameter node.

Classes

- class [ui.BoundaryFileParameterController](#)
A controller for a boundary file parameter node.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.51.1 Detailed Description

Controller for a boundary file parameter node.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

This controller can set up a view suited for file browsing.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.52 src/ui/DirectoryExtensionController.java File Reference

Controller for the directory extension name used in FullSWOF.

Classes

- class [ui.DirectoryExtensionController](#)

A controller for a directory extension parameter.

Packages

- package [ui](#)

This package provides the controllers and views associated with the model classes.

7.52.1 Detailed Description

Controller for the directory extension name used in FullSWOF.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for a directory extension parameter. The view provided is similar to that of a FieldParameterController but the validation method includes the creation of a folder.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.53 src/ui/ExternalNodeController.java File Reference

External node controller.

Classes

- class [ui.ExternalNodeController](#)

A controller for an external node.

Packages

- package [ui](#)

This package provides the controllers and views associated with the model classes.

7.53.1 Detailed Description

External node controller.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for an external node.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.54 src/ui/FieldParameterController.java File Reference

Controller of FullSWOF parameters.

Classes

- class [ui.FieldParameterController](#)
A controller for a field parameter.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.54.1 Detailed Description

Controller of FullSWOF parameters.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for a field parameter. details This controller provides a view made of a label followed by a text field for input.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.55 src/ui/FileBuilderController.java File Reference

Controller for parameters that are not in FullSWOF.

Classes

- class [ui.FileBuilderController](#)
A controller for a file builder parameter.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.55.1 Detailed Description

Controller for parameters that are not in FullSWOF.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a file builder parameter.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.56 src/ui/FileBuilderWithoutSavedController.java File Reference

Controller for parameters that are not in FullSWOF, verifying errors.

Classes

- class [ui.FileBuilderWithoutSavedController](#)
A controller for a file builder parameter.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.56.1 Detailed Description

Controller for parameters that are not in FullSWOF, verifying errors.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a file builder parameter.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.57 src/ui/FileParameterController.java File Reference

Controller for the parameter that stores a file pathname.

Classes

- class [ui.FileParameterController](#)
A controller for a file parameter node.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.57.1 Detailed Description

Controller for the parameter that stores a file pathname.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a file parameter node. This controller can set up a view suited for file browsing. If the file parameter has a visualizer defined, the view enables the user to activate it.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.58 src/ui/FormulaFileBuilderController.java File Reference

Controller for the parameter with mathematical formulas.

Classes

- class [ui.FormulaFileBuilderController](#)
A controller for a file builder using parsed formulas.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.58.1 Detailed Description

Controller for the parameter with mathematical formulas.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a file builder using parsed formulas.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.59 src/ui/InternalNodeController.java File Reference

Controller of internal node.

Classes

- class [ui.InternalNodeController](#)
The controller of an internal node.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.59.1 Detailed Description

Controller of internal node.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

The controller of an internal node. The controllers maintain a hierarchy which is parallel to that of the model. The controller of an internal node must therefore maintain a list of child controllers, similar to the list of child nodes of its node.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.60 src/ui/MultipleChoiceParameterController.java File Reference

Controller for a choice parameter (list of possible values)

Classes

- class [ui.MultipleChoiceParameterController](#)
A controller for a multiple choice parameter node.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.60.1 Detailed Description

Controller for a choice parameter (list of possible values)

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a multiple choice parameter node. The view provided by this controller is made of a label and a combo box listing the possible values.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.61 src/ui/NodeController.java File Reference

Controller of node.

Classes

- class [ui.NodeController](#)

A controller for a node, in the model-view-controller pattern.

Packages

- package [ui](#)

This package provides the controllers and views associated with the model classes.

7.61.1 Detailed Description

Controller of node.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for a node, in the model-view-controller pattern. The view is an attribute of the controller, and must be instantiated by the controller itself with a call to `setUpView()`.

See also

MVC pattern

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.62 src/ui/ParametersGroupController.java File Reference

Controller for parameters.

Classes

- class [ui.ParametersGroupController](#)
A controller for an internal node.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.62.1 Detailed Description

Controller for parameters.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for an internal node. This controller can be used for any internal node, but [ui.RootController](#) might be better suited for the root of the tree. The difference between the two is only the view provided. This class provides a simple panel view with each child node on the same panel.

See also

[ui.RootController](#)

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.63 src/ui/PointFileController.java File Reference

Controller for the parameter for point file builder node.

Classes

- class [ui.PointFileController](#)
A controller for a point file builder node.
- class [ui.PointFileController.CellRenderer](#)
An instance of this class is used to render the cells of the table in the view.
- class [ui.PointFileController.PointModel](#)
The model used by the view table.

Packages

- package [ui](#)

This package provides the controllers and views associated with the model classes.

7.63.1 Detailed Description

Controller for the parameter for point file builder node.

Author

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a point file builder node. This controller can set up a view that includes an editable table where the user can write x and y value.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.64 src/ui/RainFileController.java File Reference

Controller for the parameter for rain file.

Classes

- class [ui.RainFileController](#)

A controller for a rain file builder node.

- class [ui.RainFileController.CellRenderer](#)

An instance of this class is used to render the cells of the table in the view.

- class [ui.RainFileController.RainModel](#)

The model used by the view table.

Packages

- package [ui](#)

This package provides the controllers and views associated with the model classes.

7.64.1 Detailed Description

Controller for the parameter for rain file.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A controller for a rain file builder node.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.65 src/ui/RootController.java File Reference

Controller for a node (preferentially root node)

Classes

- class [ui.RootController](#)
A controller for an internal node, especially suited for the root of the tree.

Packages

- package [ui](#)
This package provides the controllers and views associated with the model classes.

7.65.1 Detailed Description

Controller for a node (preferentially root node)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A controller for an internal node, especially suited for the root of the tree. This class can still be used for other internal nodes, nor does the root need to use this controller. The view provided by this controller is simply better suited for the root. The view is a tabbed pane, where each child node is a tab.

See also

[ui.ParametersGroupController](#) for the other type of controller that can be used for internal nodes.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.66 src/visualization/AnimatedChart.java File Reference

Animated chart.

Classes

- class [visualization.AnimatedChart](#)
An animated chart.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.66.1 Detailed Description

Animated chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An animated chart.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.67 src/visualization/AnimatedChartContent.java File Reference

Content of an animated chart.

Classes

- class [visualization.AnimatedChartContent](#)
The content of an animated chart.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.67.1 Detailed Description

Content of an animated chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

The content of an animated chart. It is made of multiple chart content displayed one after the other in order to create an animation.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.68 src/visualization/AnimatedScene.java File Reference

3D animated chart

Classes

- class [visualization.AnimatedScene](#)

A three-dimensional animated chart.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.68.1 Detailed Description

3D animated chart

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A three-dimensional animated chart.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.69 src/visualization/AnimatedSceneContent.java File Reference

Content of a scene.

Classes

- class [visualization.AnimatedSceneContent](#)
The content of a three dimensional animated chart.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.69.1 Detailed Description

Content of a scene.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

The content of a three dimensional animated chart. This content is made of multiple 3D components, each made visible during a brief time to create an animation.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.70 src/visualization/Animation.java File Reference

Animation

Classes

- class [visualization.Animation< E >](#)

An abstract class to implement animations as Swing component.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.70.1 Detailed Description

Animation

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An abstract class to implement animations as Swing component. The swing component includes a command panel to control the animation.

Implementation note: the animation must be stopped before the JPanel or its container are disposed of, otherwise the player thread may crash the application.

Parameters

<E>	the type of image used in the animation
-----	---

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.71 src/visualization/AnimationContent.java File Reference

Content of animation.

Classes

- class [visualization.AnimationContent< E >](#)

The content of an animation.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.71.1 Detailed Description

Content of animation.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

The content of an animation.

Parameters

<E>	the type of images used by the animation
-----	--

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.72 src/visualization/Chart.java File Reference

Chart

Classes

- class [visualization.Chart](#)
A chart that can be displayed as a AWT component.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.72.1 Detailed Description

Chart

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A chart that can be displayed as a AWT component.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.73 src/visualization/ChartContent.java File Reference

Content of chart.

Classes

- class [visualization.ChartContent](#)

The content of a chart, that can include different types of data.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.73.1 Detailed Description

Content of chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

The content of a chart, that can include different types of data.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.74 src/visualization/ChartData.java File Reference

Data for chart.

Classes

- class [visualization.ChartData](#)

An element of data to be displayed in a chart.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.74.1 Detailed Description

Data for chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An element of data to be displayed in a chart.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.75 src/visualization/ChartLine.java File Reference

Line in a chart.

Classes

- class [visualization.ChartLine](#)

A line to be displayed on a chart.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.75.1 Detailed Description

Line in a chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A line to be displayed on a chart.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.76 src/visualization/ChartScatterData.java File Reference

Data for scatter chart.

Classes

- class [visualization.ChartScatterData](#)
A collection of points to be displayed in a scatter chart.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.76.1 Detailed Description

Data for scatter chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A collection of points to be displayed in a scatter chart.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.77 src/visualization/DataFileReader.java File Reference

Reads data and builds chart.

Classes

- class [visualization.DataFileReader](#)

Provides static method to read a simple data file and build a chart element from it.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.77.1 Detailed Description

Reads data and builds chart.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

Provides static method to read a simple data file and build a chart element from it.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.78 src/visualization/fs1d/FS1DFile.java File Reference

FullSWOF_1D file.

Classes

- class [visualization.fs1d.FS1DFile](#)

A FullSWOF_1D output file.

Packages

- package [visualization.fs1d](#)

This package provides the classes used to visualize FullSWOF_1D output files.

7.78.1 Detailed Description

FullSWOF_1D file.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A FullSWOF_1D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.79 src/visualization/fs1d/FS1DVisualizationPane.java File Reference

FullSWOF_1D visualization pane.

Classes

- class [visualization.fs1d.FS1DVisualizationPane](#)
A visualization pane for a FullSWOF_1D output file.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.79.1 Detailed Description

FullSWOF_1D visualization pane.

Author

Simon Robillard (2012)

Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A visualization pane for a FullSWOF_1D output file. It includes an animated chart with spatial information and charts presenting the evolution of values at the boundaries.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.80 src/visualization/fs1d/GnuplotFileReader1D.java File Reference

Gnuplot reader (1D files)

Classes

- class [visualization.fs1d.GnuplotFileReader1D](#)
A reader for Gnuplot output files generated by FullSWOF_1D.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.80.1 Detailed Description

Gnuplot reader (1D files)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A reader for Gnuplot output files generated by FullSWOF_1D.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.81 src/visualization/fs1d/HU1DVisualizer.java File Reference

Visualization of hu files.

Classes

- class [visualization.fs1d.HU1DVisualizer](#)
A tool used to get a quick visualization of water input files for FullSWOF_1D.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.81.1 Detailed Description

Visualization of hu files.

Author

Simon Robillard (2012)

Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Version

1.01.01

Date

2016-03-17

A tool used to get a quick visualization of water input files for FullSWOF_1D. The file is represented as a chart with a line for water height and another for water velocity.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.82 src/visualization/fs1d/OutputPoint1D.java File Reference

FullSWOF_1D cell.

Classes

- class [visualization.fs1d.OutputPoint1D](#)
A cell in a FullSWOF_1D output file.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.82.1 Detailed Description

FullSWOF_1D cell.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A cell in a FullSWOF_1D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.83 src/visualization/fs1d/RainFileVisualizer.java File Reference

Visualization of rain (1D)

Classes

- class [visualization.fs1d.RainFileVisualizer](#)
A tool used to get a quick visualization of rain input files for FullSWOF_1D.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.83.1 Detailed Description

Visualization of rain (1D)

Author

Simon Robillard (2012)
Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Version

1.01.01

Date

2016-03-17

A tool used to get a quick visualization of rain input files for FullSWOF_1D. The file is represented as a chart showing the evolution of rain during time.

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.84 src/visualization/fs2d/RainFileVisualizer.java File Reference

Visualization of rain (2D)

Classes

- class [visualization.fs2d.RainFileVisualizer](#)
A tool used to get a quick visualization of rain input files for FullSWOF_2D.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.84.1 Detailed Description

Visualization of rain (2D)

Author

Simon Robillard (2012)

Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Version

1.01.01

Date

2016-03-17

A tool used to get a quick visualization of rain input files for FullSWOF_2D. The file is represented as a chart showing the evolution of rain during time.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.85 src/visualization/fs1d/TimeLine1D.java File Reference

Collection of 1D times steps.

Classes

- class [visualization.fs1d.TimeLine1D](#)
A collection of [TimeStep1D](#) ordered by ascending time.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.85.1 Detailed Description

Collection of 1D times steps.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A collection of TimeStep1D ordered by ascending time.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.86 src/visualization/fs1d/TimeStep1D.java File Reference

Time step (1D)

Classes

- class [visualization.fs1d.TimeStep1D](#)
A time step in a FullSWOF_1D output file.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.86.1 Detailed Description

Time step (1D)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A time step in a FullSWOF_1D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.87 src/visualization/fs1d/Topography1DVisualizer.java File Reference

Visualization of the topography (1D)

Classes

- class [visualization.fs1d.Topography1DVisualizer](#)
A tool used to get a quick visualization of topography input files for FullSWOF_1D.

Packages

- package [visualization.fs1d](#)
This package provides the classes used to visualize FullSWOF_1D output files.

7.87.1 Detailed Description

Visualization of the topography (1D)

Author

Simon Robillard (2012)

Christian Laguerre christian.laguerre@math.cnrs.fr (2012-2016)

Version

1.01.01

Date

2016-03-17

A tool used to get a quick visualization of topography input files for FullSWOF_1D. The file is represented as a chart with a line topography height.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.88 src/visualization/fs2d/FS2DFile.java File Reference

FullSWOF_2D file.

Classes

- class [visualization.fs2d.FS2DFile](#)
A FullSWOF_2D output file.
- enum [visualization.fs2d.FS2DFile.Format](#)
The different formats of files produced by FullSWOF_2D.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.88.1 Detailed Description

FullSWOF_2D file.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A FullSWOF_2D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.89 src/visualization/fs2d/FS2DVisualizationPane.java File Reference

FullSWOF_2D visualization pane.

Classes

- class [visualization.fs2d.FS2DVisualizationPane](#)
A tabbed pane presenting a FullSWOF_2D File.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.89.1 Detailed Description

FullSWOF_2D visualization pane.

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A tabbed pane presenting a FullSWOF_2D File. The first tab is an animated spatial representation. The other four tabs show the evolution during time at the boundaries ; these tabs are activated only if the file contains multiple time steps.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.90 src/visualization/fs2d/GnuplotFileReader2D.java File Reference

Gnuplot reader (2D files)

Classes

- class [visualization.fs2d.GnuplotFileReader2D](#)

A reader for Gnuplot output files generated by FullSWOF_2D.

Packages

- package [visualization.fs2d](#)

This package provides the classes used to visualize FullSWOF_2D output files.

7.90.1 Detailed Description

Gnuplot reader (2D files)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A reader for Gnuplot output files generated by FullSWOF_2D.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.91 src/visualization/fs2d/HUV2DVisualizer.java File Reference

Visualization of huv files.

Classes

- class [visualization.fs2d.HUV2DVisualizer](#)

A tool used to get a quick visualization of water input files for FullSWOF_2D.

Packages

- package [visualization.fs2d](#)

This package provides the classes used to visualize FullSWOF_2D output files.

7.91.1 Detailed Description

Visualization of huv files.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A tool used to get a quick visualization of water input files for FullSWOF_2D. The file is represented as a 3D surface showing the water height. The color of the surface indicate the velocity of the water in each point.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.92 src/visualization/fs2d/OutputPoint2D.java File Reference

FullSWOF_2D cell.

Classes

- class [visualization.fs2d.OutputPoint2D](#)
A cell in a FullSWOF_2D output file.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.92.1 Detailed Description

FullSWOF_2D cell.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A cell in a FullSWOF_2D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.93 src/visualization/fs2d/TimeLine2D.java File Reference

Collection of 2D times steps.

Classes

- class [visualization.fs2d.TimeLine2D](#)
A collection of [TimeStep2D](#) ordered by ascending time.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.93.1 Detailed Description

Collection of 2D times steps.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A collection of TimeStep2D ordered by ascending time.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.94 src/visualization/fs2d/TimeStep2D.java File Reference

Time step (2D)

Classes

- class [visualization.fs2d.TimeStep2D](#)
A time step in a FullSWOF_2D output file.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.94.1 Detailed Description

Time step (2D)

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A time step in a FullSWOF_2D output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.95 src/visualization/fs2d/Topography2DVisualizer.java File Reference

Visualization of the topography (2D)

Classes

- class [visualization.fs2d.Topography2DVisualizer](#)
A tool used to get a quick visualization of topography input for FullSWOF_2D.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.95.1 Detailed Description

Visualization of the topography (2D)

Author

Simon Robillard (2012)
Marion Juré (2020)

Version

2.00.00

Date

2020-06-12

A tool used to get a quick visualization of topography input for FullSWOF_2D. The file is represented as a 3D surface showing topography.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.96 src/visualization/fs2d/VtkFileReader2D.java File Reference

VTK reader (2D files)

Classes

- class [visualization.fs2d.VtkFileReader2D](#)
A reader for VTK output files generated by FullSWOF_2D.

Packages

- package [visualization.fs2d](#)
This package provides the classes used to visualize FullSWOF_2D output files.

7.96.1 Detailed Description

VTK reader (2D files)

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A reader for VTK output files generated by FullSWOF_2D.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.97 src/visualization/GnuplotFileReader.java File Reference

Gnuplot reader.

Classes

- class [visualization.GnuplotFileReader< OUTPUT_POINT extends OutputPoint, TIME_STEP extends TimeStep< OUTPUT_POINT, TIME_LINE extends TimeLine< TIME_STEP >](#)
A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.97.1 Detailed Description

Gnuplot reader.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A partial implementation of a reader for Gnuplot file (FullSWOF_1D and FullSWOF_2D)

Parameters

<OUTPUT_POINT>	the type of output points created by this reader
<TIME_STEP>	the type of time steps created by this reader
<TIME_LINE>	the type of timelines created by this reader

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.98 src/visualization/InputFileVisualizer.java File Reference

Visualization of input files.

Classes

- interface [visualization.InputFileVisualizer](#)

A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.98.1 Detailed Description

Visualization of input files.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A tool used to get a quick visualization of input files (such as topography files, HUV files, rain files...)

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.99 src/visualization/JRealityViewingComponent.java File Reference

Builds a viewing component.

Classes

- class [visualization.JRealityViewingComponent](#)

This class provides a static method to build a viewing component for a JReality SceneGraphComponent.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.99.1 Detailed Description

Builds a viewing component.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

This class provides a static method to build a viewing component for a JReality SceneGraphComponent. The viewing component includes rotating, dragging and zooming tools as well as XYZ axes on the figure.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.100 src/visualization/OutputFileReader.java File Reference

Output reader.

Classes

- class [visualization.OutputFileReader](#)
An abstract class to implement readers for FullSWOF output files.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.100.1 Detailed Description

Output reader.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

An abstract class to implement readers for FullSWOF output files. A reader should be able to update a timeline according to the data in the file. It can do this with a complete file or update the timeline during modifications of the physical file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.101 src/visualization/OutputPoint.java File Reference

Output point.

Classes

- class [visualization.OutputPoint](#)
A cell in a FullSWOF output file.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.101.1 Detailed Description

Output point.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A cell in a FullSWOF output file.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.102 src/visualization/TimeLine.java File Reference

Collection of time steps.

Classes

- class [visualization.TimeLine](#) < E extends [TimeStep](#) < ? > >
A collection of time step in a FullSWOF evolution file.

Packages

- package [visualization](#)
This package provides classes used to visualize FullSWOF output files.

7.102.1 Detailed Description

Collection of time steps.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A collection of time step in a FullSWOF evolution file.

Parameters

<E>	the type time step in the collection
-----	--------------------------------------

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.103 src/visualization/TimeStep.java File Reference

Time step.

Classes

- class [visualization.TimeStep< E extends OutputPoint >](#)

A time step in a FullSWOF output file.

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.103.1 Detailed Description

Time step.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A time step in a FullSWOF output file. A time step is a collection of cells.

Parameters

<E>	the type of output points included in the time step
-----	---

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.104 src/visualization/VisualizationFile.java File Reference

Output file.

Classes

- class [visualization.VisualizationFile](#)< E extends [TimeLine](#)<?>>

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.104.1 Detailed Description

Output file.

Author

Simon Robillard (2012)
Antoine Schellenberger (2015)

Version

1.01.01

Date

2015-07-10

- A FullSWOF output file

Parameters

<E>	the type of time line associated with this file
-----	---

Copyright

License Cecill-V2
http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

7.105 src/visualization/VisualizationPane.java File Reference

Pane for the visualization.

Classes

- class [visualization.VisualizationPane](#)

A tabbed pane presenting a [VisualizationFile](#).

Packages

- package [visualization](#)

This package provides classes used to visualize FullSWOF output files.

7.105.1 Detailed Description

Pane for the visualization.

Author

Simon Robillard (2012)

Version

1.01.01

Date

2015-07-10

A tabbed pane presenting a VisualizationFile.

Copyright

License Cecill-V2

http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

(c) CNRS - Universite d'Orleans - INRA (France)

Index

- [static initializer]
 - parser::FormulaLexer, 136
- about
 - io::Procedures, 246
- actionPerformed
 - io::DataSetBuilderDialog, 84
 - io::MainFrame, 207
 - io::PreferencesDialog, 245
 - io::ProgressDialog, 255
 - ui::FileParameterController, 122
 - ui::FormulaFileBuilderController, 128
 - ui::MultipleChoiceParameterController, 213
 - visualization::Animation, 36
 - visualization::Chart, 63
 - visualization::fs1d::FS1DVisualizationPane, 175
 - visualization::fs2d::FS2DVisualizationPane, 177
- add
 - model::ListFile, 205
 - visualization::AnimatedChartContent, 29
 - visualization::AnimatedSceneContent, 34
 - visualization::AnimationContent, 42
 - visualization::Chart, 63
 - visualization::ChartContent, 71
 - visualization::ChartData, 75
 - visualization::fs1d::TimeLine1D, 277
- addChangeListener
 - model::ExternalNode, 100
 - model::InternalNode, 196
 - model::Node, 215
 - visualization::OutputFileReader, 222
- addConsoleLine
 - io::ProgressDialog, 256
- addController
 - ui::InternalNodeController, 200
- addDependency
 - model::ExternalNode, 100
 - model::FileBuilderParameter, 112
- addImage
 - visualization::AnimatedChart, 28
 - visualization::AnimatedScene, 33
 - visualization::Animation, 37
- addNode
 - model::InternalNode, 197
- addPossibleValue
 - model::MultipleChoiceParameter, 210
- AnimatedChart
 - visualization::AnimatedChart, 27
- AnimatedChartContent
 - visualization::AnimatedChartContent, 29
- AnimatedScene
 - visualization::AnimatedScene, 32
- AnimatedSceneContent
 - visualization::AnimatedSceneContent, 34
- Animation
 - visualization::Animation, 36
- AnimationContent
 - visualization::AnimationContent, 42
- atom
 - parser::FormulaParser, 154
- backgroundComponent
 - visualization::Animation, 39
- BoundaryFileController
 - ui::BoundaryFileController, 44
- BoundaryFileParameter
 - model::BoundaryFileParameter, 47
- BoundaryFileParameterController
 - ui::BoundaryFileParameterController, 51
- BoundaryModel
 - ui::BoundaryFileController::BoundaryModel, 53
- BoundaryParameter
 - model::BoundaryParameter, 56
- buildDataSet
 - visualization::DataFileReader, 83
- changeEvent
 - model::ExternalNode, 103
 - visualization::OutputFileReader, 223
- changeSinceLastSave
 - io::Procedures, 254
- Chart
 - visualization::Chart, 62
- ChartContent
 - visualization::ChartContent, 71
- ChartData
 - visualization::ChartData, 74, 75
- ChartLine
 - visualization::ChartLine, 79, 80
- ChartScatterData
 - visualization::ChartScatterData, 81, 82
- check

- model::ListFile, 205
- model::SettingDependency2, 274
- checkFile
 - model::definition::Definition_1D, 85
 - model::definition::Definition_2D, 86
- checkIsValidTable
 - model::definition::Definition_1D, 85
 - model::definition::Definition_2D, 86
- checkSaved
 - model::ListFile, 205
- clear
 - visualization::Chart, 63
 - visualization::ChartContent, 71
- close
 - io::Procedures, 246
- color
 - visualization::ChartData, 78
- commandPanel
 - visualization::Animation, 37
- compareTo
 - visualization::fs1d::OutputPoint1D, 225
 - visualization::fs2d::OutputPoint2D, 228
- configuration
 - model::definition::Definition_1D, 85
 - model::definition::Definition_2D, 86
- container
 - visualization::ChartData, 78
- content
 - visualization::Animation, 40
- controllerList
 - ui::InternalNodeController, 201
- copyFile
 - ui::FileParameterController, 122
- currentLocale
 - io::Start, 275
- currentPosition
 - visualization::Animation, 40
- DFA12
 - parser::FormulaLexer::DFA12, 90
- DFA12_accept
 - parser::FormulaLexer, 142
- DFA12_acceptS
 - parser::FormulaLexer, 142
- DFA12_eof
 - parser::FormulaLexer, 142
- DFA12_eofS
 - parser::FormulaLexer, 142
- DFA12_eot
 - parser::FormulaLexer, 142
- DFA12_eotS
 - parser::FormulaLexer, 142
- DFA12_max
 - parser::FormulaLexer, 142
- DFA12_maxS
 - parser::FormulaLexer, 142
- DFA12_min
 - parser::FormulaLexer, 142
- DFA12_minS
 - parser::FormulaLexer, 143
- DFA12_special
 - parser::FormulaLexer, 143
- DFA12_specialS
 - parser::FormulaLexer, 143
- DFA12_transition
 - parser::FormulaLexer, 143
- DFA12_transitionS
 - parser::FormulaLexer, 143
- DFA8
 - parser::FormulaLexer::DFA8, 91
- DFA8_accept
 - parser::FormulaLexer, 143
- DFA8_acceptS
 - parser::FormulaLexer, 143
- DFA8_eof
 - parser::FormulaLexer, 143
- DFA8_eofS
 - parser::FormulaLexer, 144
- DFA8_eot
 - parser::FormulaLexer, 144
- DFA8_eotS
 - parser::FormulaLexer, 144
- DFA8_max
 - parser::FormulaLexer, 144
- DFA8_maxS
 - parser::FormulaLexer, 144
- DFA8_min
 - parser::FormulaLexer, 144
- DFA8_minS
 - parser::FormulaLexer, 144
- DFA8_special
 - parser::FormulaLexer, 144
- DFA8_specialS
 - parser::FormulaLexer, 144
- DFA8_transition
 - parser::FormulaLexer, 145
- DFA8_transitionS
 - parser::FormulaLexer, 145
- data
 - visualization::ChartData, 78
- defaultLineColor
 - visualization::VisualizationPane, 302
- dependencies
 - model::ExternalNode, 104
- Dependency
 - model::Dependency, 88
- description

- model::Node, [217](#)
- dfa12
 - parser::FormulaLexer, [141](#)
- dfa8
 - parser::FormulaLexer, [143](#)
- DirectoryExtensionController
 - ui::DirectoryExtensionController, [92](#)
- DirectoryExtensionParameter
 - model::DirectoryExtensionParameter, [94](#)
- DisablingDependency
 - model::DisablingDependency, [95](#)
- dx
 - visualization::fs2d::OutputPoint2D, [230](#)
- dy
 - visualization::fs2d::OutputPoint2D, [231](#)
- EOF
 - parser::FormulaLexer, [145](#)
 - parser::FormulaParser, [156](#)
- EXPONENT
 - parser::FormulaLexer, [145](#)
 - parser::FormulaParser, [156](#)
- enabled
 - model::ExternalNode, [104](#)
- EnablingDependency
 - model::EnablingDependency, [97](#)
- equals
 - visualization::fs1d::OutputPoint1D, [225](#)
 - visualization::fs2d::OutputPoint2D, [228](#)
- exec
 - io::FullswofIO, [178](#)
- expr
 - parser::FormulaParser, [154](#)
- ExternalNode
 - model::ExternalNode, [99](#)
- ExternalNodeController
 - ui::ExternalNodeController, [105](#)
- FLOAT
 - parser::FormulaLexer, [145](#)
 - parser::FormulaParser, [156](#)
- FOLLOW_10_in_multExpr128
 - parser::FormulaParser, [156](#)
- FOLLOW_11_in_expr80
 - parser::FormulaParser, [156](#)
- FOLLOW_11_in_signedAtom489
 - parser::FormulaParser, [156](#)
- FOLLOW_12_in_function305
 - parser::FormulaParser, [156](#)
- FOLLOW_13_in_function213
 - parser::FormulaParser, [156](#)
- FOLLOW_13_in_function353
 - parser::FormulaParser, [156](#)
- FOLLOW_13_in_function368
 - parser::FormulaParser, [157](#)
- FOLLOW_13_in_function383
 - parser::FormulaParser, [157](#)
- FOLLOW_14_in_expr91
 - parser::FormulaParser, [157](#)
- FOLLOW_14_in_signedAtom501
 - parser::FormulaParser, [157](#)
- FOLLOW_15_in_multExpr139
 - parser::FormulaParser, [157](#)
- FOLLOW_16_in_function165
 - parser::FormulaParser, [157](#)
- FOLLOW_17_in_function176
 - parser::FormulaParser, [157](#)
- FOLLOW_18_in_function187
 - parser::FormulaParser, [157](#)
- FOLLOW_19_in_function198
 - parser::FormulaParser, [157](#)
- FOLLOW_20_in_function209
 - parser::FormulaParser, [158](#)
- FOLLOW_21_in_function224
 - parser::FormulaParser, [158](#)
- FOLLOW_22_in_function235
 - parser::FormulaParser, [158](#)
- FOLLOW_23_in_function246
 - parser::FormulaParser, [158](#)
- FOLLOW_24_in_function257
 - parser::FormulaParser, [158](#)
- FOLLOW_25_in_function268
 - parser::FormulaParser, [158](#)
- FOLLOW_26_in_function279
 - parser::FormulaParser, [158](#)
- FOLLOW_27_in_function290
 - parser::FormulaParser, [158](#)
- FOLLOW_28_in_function301
 - parser::FormulaParser, [158](#)
- FOLLOW_29_in_function316
 - parser::FormulaParser, [159](#)
- FOLLOW_30_in_function327
 - parser::FormulaParser, [159](#)
- FOLLOW_31_in_function338
 - parser::FormulaParser, [159](#)
- FOLLOW_32_in_function349
 - parser::FormulaParser, [159](#)
- FOLLOW_33_in_function364
 - parser::FormulaParser, [159](#)
- FOLLOW_34_in_function379
 - parser::FormulaParser, [159](#)
- FOLLOW_35_in_function394
 - parser::FormulaParser, [159](#)
- FOLLOW_36_in_function401
 - parser::FormulaParser, [159](#)
- FOLLOW_37_in_function412
 - parser::FormulaParser, [159](#)

- FOLLOW_38_in_function423
 - parser::FormulaParser, [160](#)
- FOLLOW_39_in_function434
 - parser::FormulaParser, [160](#)
- FOLLOW_40_in_function445
 - parser::FormulaParser, [160](#)
- FOLLOW_41_in_function456
 - parser::FormulaParser, [160](#)
- FOLLOW_42_in_function467
 - parser::FormulaParser, [160](#)
- FOLLOW_8_in_atom537
 - parser::FormulaParser, [160](#)
- FOLLOW_9_in_atom541
 - parser::FormulaParser, [160](#)
- FOLLOW_9_in_function169
 - parser::FormulaParser, [160](#)
- FOLLOW_9_in_function180
 - parser::FormulaParser, [160](#)
- FOLLOW_9_in_function191
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function202
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function217
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function228
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function239
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function250
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function261
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function272
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function283
 - parser::FormulaParser, [161](#)
- FOLLOW_9_in_function294
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function309
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function320
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function331
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function342
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function357
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function372
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function387
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function405
 - parser::FormulaParser, [162](#)
- FOLLOW_9_in_function416
 - parser::FormulaParser, [163](#)
- FOLLOW_9_in_function427
 - parser::FormulaParser, [163](#)
- FOLLOW_9_in_function438
 - parser::FormulaParser, [163](#)
- FOLLOW_9_in_function449
 - parser::FormulaParser, [163](#)
- FOLLOW_9_in_function460
 - parser::FormulaParser, [163](#)
- FOLLOW_9_in_function471
 - parser::FormulaParser, [163](#)
- FOLLOW_FLOAT_in_atom522
 - parser::FormulaParser, [167](#)
- FOLLOW_ID_in_atom529
 - parser::FormulaParser, [167](#)
- FOLLOW_NEWLINE_in_formula49
 - parser::FormulaParser, [168](#)
- FOLLOW_NEWLINE_in_formula56
 - parser::FormulaParser, [168](#)
- FOLLOW_atom_in_signedAtom494
 - parser::FormulaParser, [163](#)
- FOLLOW_atom_in_signedAtom505
 - parser::FormulaParser, [163](#)
- FOLLOW_expr_in_atom539
 - parser::FormulaParser, [163](#)
- FOLLOW_expr_in_formula47
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function168
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function179
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function190
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function201
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function212
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function216
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function227
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function238
 - parser::FormulaParser, [164](#)
- FOLLOW_expr_in_function249
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function260
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function271
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function282
 - parser::FormulaParser, [165](#)

- FOLLOW_expr_in_function293
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function304
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function308
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function319
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function330
 - parser::FormulaParser, [165](#)
- FOLLOW_expr_in_function341
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function352
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function356
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function367
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function371
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function382
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function386
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function404
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function415
 - parser::FormulaParser, [166](#)
- FOLLOW_expr_in_function426
 - parser::FormulaParser, [167](#)
- FOLLOW_expr_in_function437
 - parser::FormulaParser, [167](#)
- FOLLOW_expr_in_function448
 - parser::FormulaParser, [167](#)
- FOLLOW_expr_in_function459
 - parser::FormulaParser, [167](#)
- FOLLOW_expr_in_function470
 - parser::FormulaParser, [167](#)
- FOLLOW_function_in_atom548
 - parser::FormulaParser, [167](#)
- FOLLOW_multExpr_in_expr73
 - parser::FormulaParser, [167](#)
- FOLLOW_multExpr_in_expr84
 - parser::FormulaParser, [168](#)
- FOLLOW_multExpr_in_expr95
 - parser::FormulaParser, [168](#)
- FOLLOW_signedAtom_in_multExpr120
 - parser::FormulaParser, [168](#)
- FOLLOW_signedAtom_in_multExpr132
 - parser::FormulaParser, [168](#)
- FOLLOW_signedAtom_in_multExpr143
 - parser::FormulaParser, [168](#)
- FS1DFile
 - visualization::fs1d::FS1DFile, [173](#)
- FS1DVisualizationPane
 - visualization::fs1d::FS1DVisualizationPane, [174](#)
- FS2DFile
 - visualization::fs2d::FS2DFile, [176](#)
- FS2DVisualizationPane
 - visualization::fs2d::FS2DVisualizationPane, [177](#)
- FieldParameter
 - model::FieldParameter, [107](#)
- FieldParameterController
 - ui::FieldParameterController, [109](#)
- file
 - ui::BoundaryFileController::BoundaryModel, [55](#)
- FileBuilderController
 - ui::FileBuilderController, [111](#)
- FileBuilderParameter
 - model::FileBuilderParameter, [112](#)
- FileBuilderWithoutSavedController
 - ui::FileBuilderWithoutSavedController, [116](#)
- fileName
 - model::FileBuilderParameter, [115](#)
- FileParameter
 - model::FileParameter, [118](#), [119](#)
- FileParameterController
 - ui::FileParameterController, [122](#)
- fireChangeEvent
 - model::ExternalNode, [100](#)
 - visualization::OutputFileReader, [222](#)
- first
 - visualization::Animation, [40](#)
- FloatParameter
 - model::FloatParameter, [124](#), [125](#)
- focusGained
 - ui::NodeController, [218](#)
- focusLost
 - ui::ExternalNodeController, [105](#)
 - ui::NodeController, [219](#)
- formula
 - parser::FormulaParser, [154](#)
- FormulaFileBuilderController
 - ui::FormulaFileBuilderController, [128](#)
- FormulaFileBuilderParameter
 - model::FormulaFileBuilderParameter, [130](#)
- formulaLabels
 - model::FormulaFileBuilderParameter, [132](#)
- FormulaLexer
 - parser::FormulaLexer, [136](#)
- FormulaParser
 - parser::FormulaParser, [153](#)
- formulas
 - model::FormulaFileBuilderParameter, [132](#)
- frColor
 - visualization::VisualizationPane, [302](#)

- fromFile
 - model::BoundaryFileParameter, 47
 - model::BoundaryParameter, 57
 - model::ExternalNode, 101
 - model::FileBuilderParameter, 114
 - model::FileParameter, 119
 - model::InternalNode, 197
 - model::Node, 215
 - model::PointFileParameter, 236
 - model::RainFileParameter, 260
- function
 - parser::FormulaParser, 154
- GNUPLOT
 - visualization::fs2d::FS2DFile::Format, 127
- get
 - visualization::AnimationContent, 42
- getAvailableConfigurations
 - io::Procedures, 247
- getBrown
 - model::BoundaryFileParameter, 48
- getChartLineFr
 - visualization::fs1d::TimeStep1D, 284
- getChartLineHZ
 - visualization::fs1d::TimeStep1D, 285
- getChartLineH
 - visualization::fs1d::TimeStep1D, 284
- getChartLineQ
 - visualization::fs1d::TimeStep1D, 285
- getChartLineU
 - visualization::fs1d::TimeStep1D, 285
- getChartLineZ
 - visualization::fs1d::TimeStep1D, 285
- getChartTitle
 - visualization::ChartContent, 71
- getColor
 - visualization::ChartData, 75
- getColumnClass
 - ui::BoundaryFileController::BoundaryModel, 53
 - ui::PointFileController::PointModel, 239
 - ui::RainFileController::RainModel, 265
- getColumnCount
 - ui::BoundaryFileController::BoundaryModel, 53
 - ui::PointFileController::PointModel, 240
 - ui::RainFileController::RainModel, 266
- getColumnName
 - ui::BoundaryFileController::BoundaryModel, 53
 - ui::PointFileController::PointModel, 240
 - ui::RainFileController::RainModel, 266
- getContainer
 - visualization::ChartContent, 71
 - visualization::ChartData, 76
- getContent
 - visualization::Animation, 37
- visualization::Chart, 63
- getCurrentPosition
 - visualization::Animation, 37
- getData
 - visualization::ChartContent, 72
- getDefaultConfiguration
 - io::Procedures, 247
- getDelegates
 - parser::FormulaLexer, 136
 - parser::FormulaParser, 154
- getDependencies
 - model::ExternalNode, 101
- getDescription
 - parser::FormulaLexer::DFA12, 91
 - parser::FormulaLexer::DFA8, 91
- getDx
 - visualization::fs1d::OutputPoint1D, 225
 - visualization::fs2d::OutputPoint2D, 228
- getDy
 - visualization::fs2d::OutputPoint2D, 228
- getFile
 - io::Procedures, 247
 - model::BoundaryFileParameter, 48
- getFileContent
 - model::BoundaryFileParameter, 48
 - model::FileBuilderParameter, 114
 - model::FormulaFileBuilderParameter, 130
 - model::HU1DBuilderParameter, 188
 - model::HUV2DBuilderParameter, 190
 - model::PointFileParameter, 236
 - model::RainFileParameter, 260
 - model::Topography1DBuilderParameter, 292
 - model::Topography2DBuilderParameter, 294
- getFileName
 - model::FileBuilderParameter, 114
- getFormula
 - model::FormulaFileBuilderParameter, 130
- getFormulaLabel
 - model::FormulaFileBuilderParameter, 130
- getFr
 - visualization::fs1d::OutputPoint1D, 225
 - visualization::fs2d::OutputPoint2D, 229
- getFrSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 287
- getGrammarFileName
 - parser::FormulaLexer, 137
- getHSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 287
- getHZSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 288
- getHZ
 - visualization::fs1d::OutputPoint1D, 225
 - visualization::fs2d::OutputPoint2D, 229

- getImage
 - visualization::Chart, 64
- getImagesCount
 - visualization::Animation, 37
 - visualization::AnimationContent, 42
- getImagesPerSecond
 - visualization::Animation, 38
- getLastDirectoryUsed
 - io::Procedures, 247
- getLaunchingCommands
 - io::Procedures, 247
- getListX
 - model::PointFileParameter, 236
- getListY
 - model::PointFileParameter, 237
- getMaster
 - model::Dependency, 88
- getMemory
 - parser::FormulaParser, 155
- getMessages
 - io::Procedures, 248
- getModel
 - ui::NodeController, 219
- getNSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 288
- getName
 - model::Node, 215
 - visualization::ChartData, 76
 - visualization::VisualizationFile, 296
- getNodeList
 - model::InternalNode, 197
- getNumberOfFormulas
 - model::FormulaFileBuilderParameter, 131
- getOutputDirectory
 - io::Procedures, 248
- getPath
 - model::BoundaryFileParameter, 48
 - model::BoundaryParameter, 57
- getPathRedirection
 - io::Procedures, 248
- getPathRedirectionBrowse
 - io::Procedures, 248
- getPossibleValues
 - model::MultipleChoiceParameter, 210
- getPreferredSize
 - visualization::Chart, 64
- getPrintedValue
 - model::MultipleChoiceParameter::PossibleValue, 243
- getQSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 288
- getQx
 - visualization::fs2d::OutputPoint2D, 229
- getQxSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 288
- getQy
 - visualization::fs2d::OutputPoint2D, 230
- getQySceneGraphComponent
 - visualization::fs2d::TimeStep2D, 288
- getRain
 - model::RainFileParameter, 260
- getRecentFilesList
 - io::Procedures, 248
- getRowCount
 - ui::BoundaryFileController::BoundaryModel, 53
 - ui::PointFileController::PointModel, 240
 - ui::RainFileController::RainModel, 266
- getSize
 - visualization::Chart, 64
- getSlave
 - model::Dependency, 88
- getSlaveValue
 - model::SettingDependency, 272
- getTableCellRendererComponent
 - ui::BoundaryFileController::CellRenderer, 61
 - ui::PointFileController::CellRenderer, 60
 - ui::RainFileController::CellRenderer, 58
- getTaggedValue
 - model::BoundaryFileParameter, 48
 - model::PointFileParameter, 237
 - model::RainFileParameter, 260
- getTargetValue
 - model::Dependency, 88
- getTime
 - model::BoundaryFileParameter, 49
 - model::RainFileParameter, 260
- getTimeline
 - visualization::VisualizationFile, 296
- getTitle
 - visualization::Chart, 64
- getTokenNames
 - parser::FormulaParser, 155
- getUSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 289
- getVSceneGraphComponent
 - visualization::fs2d::TimeStep2D, 289
- getValue
 - model::ExternalNode, 101
 - model::FileBuilderParameter, 114
 - model::MultipleChoiceParameter::PossibleValue, 243
- getValueAt
 - ui::BoundaryFileController::BoundaryModel, 54
 - ui::PointFileController::PointModel, 240
 - ui::RainFileController::RainModel, 266
- getView

- ui::NodeController, [219](#)
 - visualization::VisualizationFile, [296](#)
- getVisualisation
 - visualization::fs1d::TimeLine1D, [277](#)
 - visualization::fs2d::TimeLine2D, [280](#)
- getVisualization
 - visualization::InputFileVisualizer, [192](#)
 - visualization::fs1d::HU1DVisualizer, [189](#)
 - visualization::fs1d::RainFileVisualizer, [263](#)
 - visualization::fs1d::TimeStep1D, [285](#)
 - visualization::fs1d::Topography1DVisualizer, [292](#)
 - visualization::fs2d::HUV2DVisualizer, [191](#)
 - visualization::fs2d::RainFileVisualizer, [264](#)
 - visualization::fs2d::TimeStep2D, [289](#)
 - visualization::fs2d::Topography2DVisualizer, [295](#)
- getVisualizer
 - model::FileParameter, [119](#)
 - model::FormulaFileBuilderParameter, [131](#)
- getWorkingController
 - io::Procedures, [248](#)
- getWorkingDirectory
 - io::Procedures, [249](#)
- getWorkingModel
 - io::Procedures, [249](#)
- getWorkingProject
 - io::Procedures, [249](#)
- getWorkingUI
 - io::Procedures, [249](#)
- getXmax
 - visualization::AnimatedChartContent, [31](#)
 - visualization::Chart, [64](#)
 - visualization::ChartContent, [72](#)
 - visualization::ChartData, [76](#)
- getXmaxCumulativeDischarge
 - visualization::fs1d::TimeLine1D, [278](#)
 - visualization::fs2d::TimeLine2D, [280](#)
- getXmaxDischarge
 - visualization::fs1d::TimeStep1D, [286](#)
 - visualization::fs2d::TimeStep2D, [289](#)
- getXmaxInstantDischarge
 - visualization::fs1d::TimeLine1D, [278](#)
 - visualization::fs2d::TimeLine2D, [280](#)
- getXmaxWaterHeight
 - visualization::fs1d::TimeLine1D, [278](#)
 - visualization::fs1d::TimeStep1D, [286](#)
 - visualization::fs2d::TimeLine2D, [280](#)
 - visualization::fs2d::TimeStep2D, [289](#)
- getXmin
 - visualization::AnimatedChartContent, [31](#)
 - visualization::Chart, [64](#)
 - visualization::ChartContent, [72](#)
 - visualization::ChartData, [76](#)
- getXminCumulativeDischarge
 - visualization::fs1d::TimeLine1D, [278](#)
 - visualization::fs2d::TimeLine2D, [280](#)
- getXminDischarge
 - visualization::fs1d::TimeStep1D, [286](#)
 - visualization::fs2d::TimeStep2D, [290](#)
- getXminInstantDischarge
 - visualization::fs1d::TimeLine1D, [278](#)
 - visualization::fs2d::TimeLine2D, [281](#)
- getXminWaterHeight
 - visualization::fs1d::TimeLine1D, [279](#)
 - visualization::fs1d::TimeStep1D, [286](#)
 - visualization::fs2d::TimeLine2D, [281](#)
 - visualization::fs2d::TimeStep2D, [290](#)
- getYmax
 - visualization::AnimatedChartContent, [31](#)
 - visualization::Chart, [65](#)
 - visualization::ChartContent, [72](#)
 - visualization::ChartData, [76](#)
- getYmaxCumulativeDischarge
 - visualization::fs2d::TimeLine2D, [281](#)
- getYmaxDischarge
 - visualization::fs2d::TimeStep2D, [290](#)
- getYmaxInstantDischarge
 - visualization::fs2d::TimeLine2D, [281](#)
- getYmaxWaterHeight
 - visualization::fs2d::TimeLine2D, [281](#)
 - visualization::fs2d::TimeStep2D, [290](#)
- getYmin
 - visualization::AnimatedChartContent, [31](#)
 - visualization::Chart, [65](#)
 - visualization::ChartContent, [72](#)
 - visualization::ChartData, [76](#)
- getYminCumulativeDischarge
 - visualization::fs2d::TimeLine2D, [282](#)
- getYminDischarge
 - visualization::fs2d::TimeStep2D, [290](#)
- getYminInstantDischarge
 - visualization::fs2d::TimeLine2D, [282](#)
- getYminWaterHeight
 - visualization::fs2d::TimeLine2D, [282](#)
 - visualization::fs2d::TimeStep2D, [290](#)
- getZSceneGraphComponent
 - visualization::fs2d::TimeStep2D, [291](#)
- getH
 - visualization::fs1d::OutputPoint1D, [225](#)
 - visualization::fs2d::OutputPoint2D, [229](#)
- getN
 - visualization::fs2d::OutputPoint2D, [229](#)
- getQ
 - visualization::fs1d::OutputPoint1D, [226](#)
 - visualization::fs2d::OutputPoint2D, [229](#)
- getT
 - visualization::TimeStep, [283](#)

- getU
 - visualization::fs1d::OutputPoint1D, 226
 - visualization::fs2d::OutputPoint2D, 230
- getV
 - visualization::fs2d::OutputPoint2D, 230
- getZ
 - visualization::fs1d::OutputPoint1D, 226
 - visualization::fs2d::OutputPoint2D, 230
- GnuplotFileReader
 - visualization::GnuplotFileReader, 180
- GnuplotFileReader1D
 - visualization::fs1d::GnuplotFileReader1D, 183
- GnuplotFileReader2D
 - visualization::fs2d::GnuplotFileReader2D, 185
- h
 - visualization::fs2d::OutputPoint2D, 231
- hColor
 - visualization::VisualizationPane, 302
- HU1DBuilderParameter
 - model::HU1DBuilderParameter, 188
- HUV2DBuilderParameter
 - model::HUV2DBuilderParameter, 190
- hasChangedSinceLastSave
 - io::Procedures, 249
- help
 - io::Procedures, 249
- hierarchyChanged
 - visualization::Animation, 38
- highlightView
 - ui::BoundaryFileController, 44
 - ui::ExternalNodeController, 106
 - ui::FieldParameterController, 109
 - ui::FormulaFileBuilderController, 128
 - ui::MultipleChoiceParameterController, 213
 - ui::PointFileController, 234
 - ui::RainFileController, 257
- HtmlAbout
 - io::HtmlAbout, 186
- HtmlFrame
 - io::HtmlFrame, 187
- hzColor
 - visualization::VisualizationPane, 302
- ID
 - parser::FormulaLexer, 145
 - parser::FormulaParser, 168
- imagesPerSecond
 - visualization::Animation, 40
- importParameters
 - io::Procedures, 250
- IntegerParameter
 - model::IntegerParameter, 193, 194
- InternalNode
 - model::InternalNode, 196
- InternalNodeController
 - ui::InternalNodeController, 200
- Interval
 - model::Interval, 201, 202
- io, 19
 - io.DataSetBuilderDialog, 83
 - io.FullswofIO, 178
 - io.HtmlAbout, 186
 - io.HtmlFrame, 186
 - io.MainFrame, 206
 - io.PreferencesDialog, 244
 - io.Procedures, 245
 - io.ProgressDialog, 255
 - io.Start, 275
 - io.VisualizationFrame, 299
 - io::DataSetBuilderDialog
 - actionPerformed, 84
 - MIN_FIELD_WIDTH, 85
 - MIN_LABEL_WIDTH, 85
 - showDialog, 84
 - io::FullswofIO
 - exec, 178
 - setSimultaneousVisualization, 178
 - simultaneousVisualization, 179
 - io::HtmlAbout
 - HtmlAbout, 186
 - io::HtmlFrame
 - HtmlFrame, 187
 - io::MainFrame
 - actionPerformed, 207
 - MainFrame, 206
 - stateChanged, 207
 - updateContent, 207
 - updateRecentFilesMenu, 207
 - windowActivated, 207
 - windowClosed, 207
 - windowClosing, 208
 - windowDeactivated, 208
 - windowDeiconified, 208
 - windowIconified, 208
 - windowOpened, 208
 - io::PreferencesDialog
 - actionPerformed, 245
 - PreferencesDialog, 244
 - io::Procedures
 - about, 246
 - changeSinceLastSave, 254
 - close, 246
 - getAvailableConfigurations, 247
 - getDefaultConfiguration, 247
 - getFile, 247
 - getLastDirectoryUsed, 247

- getLaunchingCommands, 247
- getMessages, 248
- getOutputDirectory, 248
- getPathRedirection, 248
- getPathRedirectionBrowse, 248
- getRecentFilesList, 248
- getWorkingController, 248
- getWorkingDirectory, 249
- getWorkingModel, 249
- getWorkingProject, 249
- getWorkingUI, 249
- hasChangedSinceLastSave, 249
- help, 249
- importParameters, 250
- isFile, 250
- messages, 254
- newProject, 250
- newProjectAfterConfirmation, 250
- openProject, 250, 251
- openVisualisation, 251
- projectIsReady, 251
- runProject, 251
- saveProject, 251
- saveProjectAs, 251
- setDefaultConfiguration, 252
- setLastDirectoryUsed, 252
- setLaunchingCommands, 252
- setModel, 252
- setOutputDirectory, 253
- setPathRedirection, 253
- setPathRedirectionBrowse, 253
- setPreferences, 253
- setRecentFilesList, 254
- setVerboseParametersFiles, 254
- workingModel, 254
- io::ProgressDialog
 - actionPerformed, 255
 - addConsoleLine, 256
 - ProgressDialog, 255
 - updateProgress, 256
- io::Start
 - currentLocale, 275
 - main, 275
- io::VisualizationFrame
 - VisualizationFrame, 299
 - windowActivated, 299
 - windowClosed, 299
 - windowClosing, 300
 - windowDeactivated, 300
 - windowDeiconified, 300
 - windowIconified, 300
 - windowOpened, 300
- isCellEditable
 - ui::BoundaryFileController::BoundaryModel, 54
 - ui::PointFileController::PointModel, 241
 - ui::RainFileController::RainModel, 267
- isEnabled
 - model::ExternalNode, 101
- isFile
 - io::Procedures, 250
- isFormulaValid
 - model::FormulaFileBuilderParameter, 131
- isIncluded
 - model::Interval, 203
- isPlaying
 - visualization::Animation, 38
- isRespected
 - model::Dependency, 89
 - model::DisablingDependency, 96
 - model::EnablingDependency, 97
 - model::SettingDependency, 273
- isValid
 - model::BoundaryFileParameter, 49
 - model::DirectoryExtensionParameter, 94
 - model::ExternalNode, 102
 - model::FieldParameter, 107
 - model::FileParameter, 120
 - model::FloatParameter, 126
 - model::FormulaFileBuilderParameter, 131
 - model::IntegerParameter, 194
 - model::InternalNode, 198
 - model::MultipleChoiceParameter, 211
 - model::Node, 216
 - model::PointFileParameter, 237
 - model::RainFileParameter, 261
- isValidCell
 - ui::BoundaryFileController::BoundaryModel, 54
 - ui::PointFileController::PointModel, 241
 - ui::RainFileController::RainModel, 267
- isValidTable
 - model::BoundaryFileParameter, 49
 - model::FileBuilderParameter, 114
 - model::PointFileParameter, 237
 - model::RainFileParameter, 261
- LIGHT_RED
 - ui::NodeController, 220
- last
 - visualization::Animation, 40
- ListFile
 - model::ListFile, 205
- listenerList
 - model::ExternalNode, 104
 - visualization::OutputFileReader, 223
- listX
 - model::PointFileParameter, 238
 - ui::PointFileController::PointModel, 242

- listY
 - model::PointFileParameter, 238
 - ui::PointFileController::PointModel, 242
- loadIcons
 - visualization::Animation, 38
- mEXPONENT
 - parser::FormulaLexer, 137
- mFLOAT
 - parser::FormulaLexer, 137
- MIN_BOX_WIDTH
 - ui::NodeController, 220
- MIN_FIELD_WIDTH
 - io::DataSetBuilderDialog, 85
 - ui::NodeController, 220
- MIN_LABEL_WIDTH
 - io::DataSetBuilderDialog, 85
 - ui::NodeController, 220
- mID
 - parser::FormulaLexer, 137
- mNEWLINE
 - parser::FormulaLexer, 137
- mT__10
 - parser::FormulaLexer, 137
- mT__11
 - parser::FormulaLexer, 137
- mT__12
 - parser::FormulaLexer, 137
- mT__13
 - parser::FormulaLexer, 137
- mT__14
 - parser::FormulaLexer, 138
- mT__15
 - parser::FormulaLexer, 138
- mT__16
 - parser::FormulaLexer, 138
- mT__17
 - parser::FormulaLexer, 138
- mT__18
 - parser::FormulaLexer, 138
- mT__19
 - parser::FormulaLexer, 138
- mT__20
 - parser::FormulaLexer, 138
- mT__21
 - parser::FormulaLexer, 138
- mT__22
 - parser::FormulaLexer, 138
- mT__23
 - parser::FormulaLexer, 139
- mT__24
 - parser::FormulaLexer, 139
- mT__25
 - parser::FormulaLexer, 139
- mT__26
 - parser::FormulaLexer, 139
- mT__27
 - parser::FormulaLexer, 139
- mT__28
 - parser::FormulaLexer, 139
- mT__29
 - parser::FormulaLexer, 139
- mT__30
 - parser::FormulaLexer, 139
- mT__31
 - parser::FormulaLexer, 139
- mT__32
 - parser::FormulaLexer, 140
- mT__33
 - parser::FormulaLexer, 140
- mT__34
 - parser::FormulaLexer, 140
- mT__35
 - parser::FormulaLexer, 140
- mT__36
 - parser::FormulaLexer, 140
- mT__37
 - parser::FormulaLexer, 140
- mT__38
 - parser::FormulaLexer, 140
- mT__39
 - parser::FormulaLexer, 140
- mT__40
 - parser::FormulaLexer, 140
- mT__41
 - parser::FormulaLexer, 141
- mT__42
 - parser::FormulaLexer, 141
- mT__8
 - parser::FormulaLexer, 141
- mT__9
 - parser::FormulaLexer, 141
- mTokens
 - parser::FormulaLexer, 141
- main
 - io::Start, 275
- MainFrame
 - io::MainFrame, 206
- makeBackgroundComponent
 - visualization::AnimatedChart, 28
 - visualization::AnimatedScene, 33
 - visualization::Animation, 38
- makePoint
 - visualization::GnuplotFileReader, 180
 - visualization::fs1d::GnuplotFileReader1D, 183
 - visualization::fs2d::GnuplotFileReader2D, 185
- makeTimeStep

- visualization::GnuplotFileReader, 180
- visualization::fs1d::GnuplotFileReader1D, 184
- visualization::fs2d::GnuplotFileReader2D, 185
- makeViewingComponent
 - visualization::JRealityViewingComponent, 204
- master
 - model::Dependency, 90
- messages
 - io::Procedures, 254
 - ui::FileParameterController, 123
- model, 19
 - ui::NodeController, 220
- model.BoundaryFileParameter, 46
- model.BoundaryParameter, 56
- model.definition, 21
- model.definition.Definition_1D, 85
- model.definition.Definition_2D, 86
- model.Dependency, 87
- model.DirectoryExtensionParameter, 93
- model.DisablingDependency, 95
- model.EnablingDependency, 96
- model.ExternalNode, 98
- model.FieldParameter, 106
- model.FileBuilderParameter, 111
- model.FileParameter, 117
- model.FloatParameter, 124
- model.FormulaFileBuilderParameter, 129
- model.HU1DBuilderParameter, 187
- model.HUV2DBuilderParameter, 189
- model.IntegerParameter, 192
- model.InternalNode, 195
- model.Interval, 201
- model.ListFile, 204
- model.MultipleChoiceParameter, 208
- model.MultipleChoiceParameter.PossibleValue, 242
- model.Node, 213
- model.PointFileParameter, 235
- model.RainFileParameter, 258
- model.RootNode, 269
- model.SettingDependency, 271
- model.SettingDependency2, 273
- model.Topography1DBuilderParameter, 291
- model.Topography2DBuilderParameter, 293
- model::BoundaryFileParameter
 - BoundaryFileParameter, 47
 - fromFile, 47
 - getBrown, 48
 - getFile, 48
 - getFileContent, 48
 - getPath, 48
 - getTaggedValue, 48
 - getTime, 49
 - isValid, 49
 - isValidTable, 49
 - readBoundaryFile, 49
 - setBrown, 49
 - setFile, 50
 - setTime, 50
 - setUpController, 50
- model::BoundaryParameter
 - BoundaryParameter, 56
 - fromFile, 57
 - getPath, 57
 - path, 58
 - setPath, 57
 - setUpController, 57
- model::Dependency
 - Dependency, 88
 - getMaster, 88
 - getSlave, 88
 - getTargetValue, 88
 - isRespected, 89
 - master, 90
 - resolve, 89
 - setEnabled, 89
 - slave, 90
 - stateChanged, 89
 - targetValue, 90
- model::DirectoryExtensionParameter
 - DirectoryExtensionParameter, 94
 - isValid, 94
 - setUpController, 95
- model::DisablingDependency
 - DisablingDependency, 95
 - isRespected, 96
 - resolve, 96
 - toString, 96
- model::EnablingDependency
 - EnablingDependency, 97
 - isRespected, 97
 - resolve, 98
 - toString, 98
- model::ExternalNode
 - addChangeListener, 100
 - addDependency, 100
 - changeEvent, 103
 - dependencies, 104
 - enabled, 104
 - ExternalNode, 99
 - fireChangeEvent, 100
 - fromFile, 101
 - getDependencies, 101
 - getValue, 101
 - isEnabled, 101
 - isValid, 102
 - listenerList, 104

- removeChangeListener, 102
- removeDependency, 102
- setEnabled, 102
- setValue, 102
- tag, 104
- toFile, 103
- updateChangeEvent, 103
- value, 104
- model::FieldParameter
 - FieldParameter, 107
 - isValid, 107
 - setUpController, 108
- model::FileBuilderParameter
 - addDependency, 112
 - FileBuilderParameter, 112
 - fileName, 115
 - fromFile, 114
 - getFileContent, 114
 - getFileName, 114
 - getValue, 114
 - isValidTable, 114
 - setValue, 115
 - toFile, 115
- model::FileParameter
 - FileParameter, 118, 119
 - fromFile, 119
 - getVisualizer, 119
 - isValid, 120
 - setUpController, 120
 - setVisualizer, 120
 - toFile, 120
 - visualizer, 121
- model::FloatParameter
 - FloatParameter, 124, 125
 - isValid, 126
 - setUpController, 126
- model::FormulaFileBuilderParameter
 - FormulaFileBuilderParameter, 130
 - formulaLabels, 132
 - formulas, 132
 - getFileContent, 130
 - getFormula, 130
 - getFormulaLabel, 130
 - getNumberOfFormulas, 131
 - getVisualizer, 131
 - isFormulaValid, 131
 - isValid, 131
 - parseFormula, 131
 - setFormula, 132
 - setUpController, 132
 - visualizer, 132
- model::HU1DBuilderParameter
 - getFileContent, 188
 - HU1DBuilderParameter, 188
- model::HUV2DBuilderParameter
 - getFileContent, 190
 - HUV2DBuilderParameter, 190
- model::IntegerParameter
 - IntegerParameter, 193, 194
 - isValid, 194
 - setUpController, 194
- model::InternalNode
 - addChangeListener, 196
 - addNode, 197
 - fromFile, 197
 - getNodeList, 197
 - InternalNode, 196
 - isValid, 198
 - nodeList, 199
 - setUpController, 198
 - toFile, 198
 - updateChangeEvent, 198
- model::Interval
 - Interval, 201, 202
 - isIncluded, 203
- model::ListFile
 - add, 205
 - check, 205
 - checkSaved, 205
 - ListFile, 205
- model::MultipleChoiceParameter
 - addPossibleValue, 210
 - getPossibleValues, 210
 - isValid, 211
 - MultipleChoiceParameter, 209
 - setUpController, 211
 - setValue, 211
 - toFile, 211
- model::MultipleChoiceParameter::PossibleValue
 - getPrintedValue, 243
 - getValue, 243
 - PossibleValue, 243
 - toString, 244
- model::Node
 - addChangeListener, 215
 - description, 217
 - fromFile, 215
 - getName, 215
 - isValid, 216
 - name, 217
 - Node, 214
 - setUpController, 216
 - toFile, 216
 - toString, 216
 - updateChangeEvent, 216
- model::PointFileParameter

- fromFile, [236](#)
 - getFileContent, [236](#)
 - getListX, [236](#)
 - getListY, [237](#)
 - getTaggedValue, [237](#)
 - isValid, [237](#)
 - isValidTable, [237](#)
 - listX, [238](#)
 - listY, [238](#)
 - PointFileParameter, [236](#)
 - pointFileTag, [238](#)
 - pointTag, [238](#)
 - pointTagValue, [238](#)
 - setUpController, [238](#)
- model::RainFileParameter
 - fromFile, [260](#)
 - getFileContent, [260](#)
 - getRain, [260](#)
 - getTaggedValue, [260](#)
 - getTime, [260](#)
 - isValid, [261](#)
 - isValidTable, [261](#)
 - rain, [262](#)
 - RainFileParameter, [259](#)
 - rainFileTag, [262](#)
 - rainTag, [262](#)
 - rainTagValue, [262](#)
 - setRain, [261](#)
 - setTime, [261](#)
 - setUpController, [262](#)
 - time, [262](#)
- model::RootNode
 - RootNode, [270](#)
 - setUpController, [271](#)
- model::SettingDependency
 - getSlaveValue, [272](#)
 - isRespected, [273](#)
 - resolve, [273](#)
 - SettingDependency, [272](#)
 - slaveValue, [273](#)
- model::SettingDependency2
 - check, [274](#)
 - resolve, [274](#)
 - SettingDependency2, [274](#)
- model::Topography1DBuilderParameter
 - getFileContent, [292](#)
 - Topography1DBuilderParameter, [291](#)
- model::Topography2DBuilderParameter
 - getFileContent, [294](#)
 - Topography2DBuilderParameter, [294](#)
- model::definition::Definition_1D
 - checkFile, [85](#)
 - checkIsValidTable, [85](#)
 - configuration, [85](#)
- model::definition::Definition_2D
 - checkFile, [86](#)
 - checkIsValidTable, [86](#)
 - configuration, [86](#)
 - updateChangeEvent, [86](#)
- mouseClicked
 - ui::BoundaryFileController, [44](#)
 - visualization::Chart, [65](#)
- mouseEntered
 - ui::BoundaryFileController, [45](#)
 - visualization::Chart, [65](#)
- mouseExited
 - ui::BoundaryFileController, [45](#)
 - visualization::Chart, [65](#)
- mousePressed
 - ui::BoundaryFileController, [45](#)
 - visualization::Chart, [65](#)
- mouseReleased
 - ui::BoundaryFileController, [45](#)
 - visualization::Chart, [66](#)
- multExpr
 - parser::FormulaParser, [155](#)
- MultipleChoiceParameter
 - model::MultipleChoiceParameter, [209](#)
- MultipleChoiceParameterController
 - ui::MultipleChoiceParameterController, [212](#)
- nColor
 - visualization::VisualizationPane, [303](#)
- NEWLINE
 - parser::FormulaLexer, [145](#)
 - parser::FormulaParser, [168](#)
- name
 - model::Node, [217](#)
 - visualization::ChartData, [79](#)
 - visualization::VisualizationFile, [298](#)
- newProject
 - io::Procedures, [250](#)
- newProjectAfterConfirmation
 - io::Procedures, [250](#)
- next
 - visualization::Animation, [40](#)
- Node
 - model::Node, [214](#)
- NodeController
 - ui::NodeController, [218](#)
- nodeList
 - model::InternalNode, [199](#)
- open
 - visualization::VisualizationFile, [297](#)
- openProject
 - io::Procedures, [250](#), [251](#)

- openVisualisation
 - io::Procedures, [251](#)
- OutputPoint1D
 - visualization::fs1d::OutputPoint1D, [224](#)
- OutputPoint2D
 - visualization::fs2d::OutputPoint2D, [227](#)
- paint
 - visualization::Chart, [66](#)
 - visualization::ChartContent, [72](#)
 - visualization::ChartData, [77](#)
 - visualization::ChartLine, [80](#)
 - visualization::ChartScatterData, [82](#)
- paintLegendElement
 - visualization::ChartData, [77](#)
 - visualization::ChartLine, [80](#)
 - visualization::ChartScatterData, [82](#)
- ParametersGroupController
 - ui::ParametersGroupController, [232](#)
- parseFormula
 - model::FormulaFileBuilderParameter, [131](#)
- parser, [22](#)
- parser.FormulaLexer, [133](#)
- parser.FormulaLexer.DFA12, [90](#)
- parser.FormulaLexer.DFA8, [91](#)
- parser.FormulaParser, [149](#)
- parser::FormulaLexer
 - [static initializer], [136](#)
 - DFA12_accept, [142](#)
 - DFA12_acceptS, [142](#)
 - DFA12_eof, [142](#)
 - DFA12_eofS, [142](#)
 - DFA12_eot, [142](#)
 - DFA12_eotS, [142](#)
 - DFA12_max, [142](#)
 - DFA12_maxS, [142](#)
 - DFA12_min, [142](#)
 - DFA12_minS, [143](#)
 - DFA12_special, [143](#)
 - DFA12_specialS, [143](#)
 - DFA12_transition, [143](#)
 - DFA12_transitionS, [143](#)
 - DFA8_accept, [143](#)
 - DFA8_acceptS, [143](#)
 - DFA8_eof, [143](#)
 - DFA8_eofS, [144](#)
 - DFA8_eot, [144](#)
 - DFA8_eotS, [144](#)
 - DFA8_max, [144](#)
 - DFA8_maxS, [144](#)
 - DFA8_min, [144](#)
 - DFA8_minS, [144](#)
 - DFA8_special, [144](#)
 - DFA8_specialS, [144](#)
 - DFA8_transition, [145](#)
 - DFA8_transitionS, [145](#)
 - dfa12, [141](#)
 - dfa8, [143](#)
 - EOF, [145](#)
 - EXPONENT, [145](#)
 - FLOAT, [145](#)
 - FormulaLexer, [136](#)
 - getDelegates, [136](#)
 - getGrammarFileName, [137](#)
 - ID, [145](#)
 - mEXPONENT, [137](#)
 - mFLOAT, [137](#)
 - mID, [137](#)
 - mNEWLINE, [137](#)
 - mT__10, [137](#)
 - mT__11, [137](#)
 - mT__12, [137](#)
 - mT__13, [137](#)
 - mT__14, [138](#)
 - mT__15, [138](#)
 - mT__16, [138](#)
 - mT__17, [138](#)
 - mT__18, [138](#)
 - mT__19, [138](#)
 - mT__20, [138](#)
 - mT__21, [138](#)
 - mT__22, [138](#)
 - mT__23, [139](#)
 - mT__24, [139](#)
 - mT__25, [139](#)
 - mT__26, [139](#)
 - mT__27, [139](#)
 - mT__28, [139](#)
 - mT__29, [139](#)
 - mT__30, [139](#)
 - mT__31, [139](#)
 - mT__32, [140](#)
 - mT__33, [140](#)
 - mT__34, [140](#)
 - mT__35, [140](#)
 - mT__36, [140](#)
 - mT__37, [140](#)
 - mT__38, [140](#)
 - mT__39, [140](#)
 - mT__40, [140](#)
 - mT__41, [141](#)
 - mT__42, [141](#)
 - mT__8, [141](#)
 - mT__9, [141](#)
 - mTokens, [141](#)
 - NEWLINE, [145](#)
 - recover, [141](#)

- reportError, [141](#)
- T__10, [145](#)
- T__11, [145](#)
- T__12, [146](#)
- T__13, [146](#)
- T__14, [146](#)
- T__15, [146](#)
- T__16, [146](#)
- T__17, [146](#)
- T__18, [146](#)
- T__19, [146](#)
- T__20, [146](#)
- T__21, [147](#)
- T__22, [147](#)
- T__23, [147](#)
- T__24, [147](#)
- T__25, [147](#)
- T__26, [147](#)
- T__27, [147](#)
- T__28, [147](#)
- T__29, [147](#)
- T__30, [148](#)
- T__31, [148](#)
- T__32, [148](#)
- T__33, [148](#)
- T__34, [148](#)
- T__35, [148](#)
- T__36, [148](#)
- T__37, [148](#)
- T__38, [148](#)
- T__39, [149](#)
- T__40, [149](#)
- T__41, [149](#)
- T__42, [149](#)
- T__8, [149](#)
- T__9, [149](#)
- parser::FormulaLexer::DFA12
 - DFA12, [90](#)
 - getDescription, [91](#)
- parser::FormulaLexer::DFA8
 - DFA8, [91](#)
 - getDescription, [91](#)
- parser::FormulaParser
 - atom, [154](#)
 - EOF, [156](#)
 - EXPONENT, [156](#)
 - expr, [154](#)
 - FLOAT, [156](#)
 - FOLLOW_10_in_multExpr128, [156](#)
 - FOLLOW_11_in_expr80, [156](#)
 - FOLLOW_11_in_signedAtom489, [156](#)
 - FOLLOW_12_in_function305, [156](#)
 - FOLLOW_13_in_function213, [156](#)
 - FOLLOW_13_in_function353, [156](#)
 - FOLLOW_13_in_function368, [157](#)
 - FOLLOW_13_in_function383, [157](#)
 - FOLLOW_14_in_expr91, [157](#)
 - FOLLOW_14_in_signedAtom501, [157](#)
 - FOLLOW_15_in_multExpr139, [157](#)
 - FOLLOW_16_in_function165, [157](#)
 - FOLLOW_17_in_function176, [157](#)
 - FOLLOW_18_in_function187, [157](#)
 - FOLLOW_19_in_function198, [157](#)
 - FOLLOW_20_in_function209, [158](#)
 - FOLLOW_21_in_function224, [158](#)
 - FOLLOW_22_in_function235, [158](#)
 - FOLLOW_23_in_function246, [158](#)
 - FOLLOW_24_in_function257, [158](#)
 - FOLLOW_25_in_function268, [158](#)
 - FOLLOW_26_in_function279, [158](#)
 - FOLLOW_27_in_function290, [158](#)
 - FOLLOW_28_in_function301, [158](#)
 - FOLLOW_29_in_function316, [159](#)
 - FOLLOW_30_in_function327, [159](#)
 - FOLLOW_31_in_function338, [159](#)
 - FOLLOW_32_in_function349, [159](#)
 - FOLLOW_33_in_function364, [159](#)
 - FOLLOW_34_in_function379, [159](#)
 - FOLLOW_35_in_function394, [159](#)
 - FOLLOW_36_in_function401, [159](#)
 - FOLLOW_37_in_function412, [159](#)
 - FOLLOW_38_in_function423, [160](#)
 - FOLLOW_39_in_function434, [160](#)
 - FOLLOW_40_in_function445, [160](#)
 - FOLLOW_41_in_function456, [160](#)
 - FOLLOW_42_in_function467, [160](#)
 - FOLLOW_8_in_atom537, [160](#)
 - FOLLOW_9_in_atom541, [160](#)
 - FOLLOW_9_in_function169, [160](#)
 - FOLLOW_9_in_function180, [160](#)
 - FOLLOW_9_in_function191, [161](#)
 - FOLLOW_9_in_function202, [161](#)
 - FOLLOW_9_in_function217, [161](#)
 - FOLLOW_9_in_function228, [161](#)
 - FOLLOW_9_in_function239, [161](#)
 - FOLLOW_9_in_function250, [161](#)
 - FOLLOW_9_in_function261, [161](#)
 - FOLLOW_9_in_function272, [161](#)
 - FOLLOW_9_in_function283, [161](#)
 - FOLLOW_9_in_function294, [162](#)
 - FOLLOW_9_in_function309, [162](#)
 - FOLLOW_9_in_function320, [162](#)
 - FOLLOW_9_in_function331, [162](#)
 - FOLLOW_9_in_function342, [162](#)
 - FOLLOW_9_in_function357, [162](#)
 - FOLLOW_9_in_function372, [162](#)

- [FOLLOW_9_in_function387](#), [162](#)
- [FOLLOW_9_in_function405](#), [162](#)
- [FOLLOW_9_in_function416](#), [163](#)
- [FOLLOW_9_in_function427](#), [163](#)
- [FOLLOW_9_in_function438](#), [163](#)
- [FOLLOW_9_in_function449](#), [163](#)
- [FOLLOW_9_in_function460](#), [163](#)
- [FOLLOW_9_in_function471](#), [163](#)
- [FOLLOW_FLOAT_in_atom522](#), [167](#)
- [FOLLOW_ID_in_atom529](#), [167](#)
- [FOLLOW_NEWLINE_in_formula49](#), [168](#)
- [FOLLOW_NEWLINE_in_formula56](#), [168](#)
- [FOLLOW_atom_in_signedAtom494](#), [163](#)
- [FOLLOW_atom_in_signedAtom505](#), [163](#)
- [FOLLOW_expr_in_atom539](#), [163](#)
- [FOLLOW_expr_in_formula47](#), [164](#)
- [FOLLOW_expr_in_function168](#), [164](#)
- [FOLLOW_expr_in_function179](#), [164](#)
- [FOLLOW_expr_in_function190](#), [164](#)
- [FOLLOW_expr_in_function201](#), [164](#)
- [FOLLOW_expr_in_function212](#), [164](#)
- [FOLLOW_expr_in_function216](#), [164](#)
- [FOLLOW_expr_in_function227](#), [164](#)
- [FOLLOW_expr_in_function238](#), [164](#)
- [FOLLOW_expr_in_function249](#), [165](#)
- [FOLLOW_expr_in_function260](#), [165](#)
- [FOLLOW_expr_in_function271](#), [165](#)
- [FOLLOW_expr_in_function282](#), [165](#)
- [FOLLOW_expr_in_function293](#), [165](#)
- [FOLLOW_expr_in_function304](#), [165](#)
- [FOLLOW_expr_in_function308](#), [165](#)
- [FOLLOW_expr_in_function319](#), [165](#)
- [FOLLOW_expr_in_function330](#), [165](#)
- [FOLLOW_expr_in_function341](#), [166](#)
- [FOLLOW_expr_in_function352](#), [166](#)
- [FOLLOW_expr_in_function356](#), [166](#)
- [FOLLOW_expr_in_function367](#), [166](#)
- [FOLLOW_expr_in_function371](#), [166](#)
- [FOLLOW_expr_in_function382](#), [166](#)
- [FOLLOW_expr_in_function386](#), [166](#)
- [FOLLOW_expr_in_function404](#), [166](#)
- [FOLLOW_expr_in_function415](#), [166](#)
- [FOLLOW_expr_in_function426](#), [167](#)
- [FOLLOW_expr_in_function437](#), [167](#)
- [FOLLOW_expr_in_function448](#), [167](#)
- [FOLLOW_expr_in_function459](#), [167](#)
- [FOLLOW_expr_in_function470](#), [167](#)
- [FOLLOW_function_in_atom548](#), [167](#)
- [FOLLOW_multExpr_in_expr73](#), [167](#)
- [FOLLOW_multExpr_in_expr84](#), [168](#)
- [FOLLOW_multExpr_in_expr95](#), [168](#)
- [FOLLOW_signedAtom_in_multExpr120](#), [168](#)
- [FOLLOW_signedAtom_in_multExpr132](#), [168](#)
- [FOLLOW_signedAtom_in_multExpr143](#), [168](#)
- [formula](#), [154](#)
- [FormulaParser](#), [153](#)
- [function](#), [154](#)
- [getDelegates](#), [154](#)
- [getMemory](#), [155](#)
- [getTokenNames](#), [155](#)
- [ID](#), [168](#)
- [multExpr](#), [155](#)
- [NEWLINE](#), [168](#)
- [recoverFromMismatchedSet](#), [155](#)
- [recoverFromMismatchedToken](#), [155](#)
- [signedAtom](#), [155](#)
- [T__10](#), [169](#)
- [T__11](#), [169](#)
- [T__12](#), [169](#)
- [T__13](#), [169](#)
- [T__14](#), [169](#)
- [T__15](#), [169](#)
- [T__16](#), [169](#)
- [T__17](#), [169](#)
- [T__18](#), [169](#)
- [T__19](#), [170](#)
- [T__20](#), [170](#)
- [T__21](#), [170](#)
- [T__22](#), [170](#)
- [T__23](#), [170](#)
- [T__24](#), [170](#)
- [T__25](#), [170](#)
- [T__26](#), [170](#)
- [T__27](#), [170](#)
- [T__28](#), [171](#)
- [T__29](#), [171](#)
- [T__30](#), [171](#)
- [T__31](#), [171](#)
- [T__32](#), [171](#)
- [T__33](#), [171](#)
- [T__34](#), [171](#)
- [T__35](#), [171](#)
- [T__36](#), [171](#)
- [T__37](#), [172](#)
- [T__38](#), [172](#)
- [T__39](#), [172](#)
- [T__40](#), [172](#)
- [T__41](#), [172](#)
- [T__42](#), [172](#)
- [T__8](#), [172](#)
- [T__9](#), [172](#)
- [tokenNames](#), [172](#)
- [path](#)
 - [model::BoundaryParameter](#), [58](#)
- [pause](#)
 - [visualization::Animation](#), [40](#)

- play
 - visualization::Animation, 40
- playOrPause
 - visualization::Animation, 39
- playOrPauseButton
 - visualization::Animation, 41
- playerThread
 - visualization::Animation, 41
- playing
 - visualization::Animation, 41
- PointFileController
 - ui::PointFileController, 234
- PointFileParameter
 - model::PointFileParameter, 236
- pointFileTag
 - model::PointFileParameter, 238
- PointModel
 - ui::PointFileController::PointModel, 239
- pointTag
 - model::PointFileParameter, 238
- pointTagValue
 - model::PointFileParameter, 238
- PossibleValue
 - model::MultipleChoiceParameter::PossibleValue, 243
- PreferencesDialog
 - io::PreferencesDialog, 244
- previous
 - visualization::Animation, 41
- ProgressDialog
 - io::ProgressDialog, 255
- projectIsReady
 - io::Procedures, 251
- qColor
 - visualization::VisualizationPane, 303
- qxColor
 - visualization::VisualizationPane, 303
- qyColor
 - visualization::VisualizationPane, 303
- rain
 - model::RainFileParameter, 262
 - ui::RainFileController::RainModel, 268
- RainFileController
 - ui::RainFileController, 257
- RainFileParameter
 - model::RainFileParameter, 259
- rainFileTag
 - model::RainFileParameter, 262
- RainModel
 - ui::RainFileController::RainModel, 265
- rainTag
 - model::RainFileParameter, 262
- rainTagValue
 - model::RainFileParameter, 262
- readBoundaryFile
 - model::BoundaryFileParameter, 49
- readTimeStep
 - visualization::GnuplotFileReader, 181
- reader
 - visualization::VisualizationFile, 298
- recover
 - parser::FormulaLexer, 141
- recoverFromMismatchedSet
 - parser::FormulaParser, 155
- recoverFromMismatchedToken
 - parser::FormulaParser, 155
- remove
 - visualization::ChartContent, 73
- removeChangeListener
 - model::ExternalNode, 102
 - visualization::OutputFileReader, 222
- removeDependency
 - model::ExternalNode, 102
- removeImages
 - visualization::AnimationContent, 43
- reportError
 - parser::FormulaLexer, 141
- resizeViewComponent
 - ui::NodeController, 219
- resolve
 - model::Dependency, 89
 - model::DisablingDependency, 96
 - model::EnablingDependency, 98
 - model::SettingDependency, 273
 - model::SettingDependency2, 274
- RootController
 - ui::RootController, 269
- RootNode
 - model::RootNode, 270
- runProject
 - io::Procedures, 251
- savelImage
 - visualization::Chart, 66
- saveProject
 - io::Procedures, 251
- saveProjectAs
 - io::Procedures, 251
- scaledLine
 - visualization::Chart, 66
- scaledPoint
 - visualization::Chart, 67
- setBoundaries
 - visualization::Chart, 67
- setBoundariesToInclude
 - visualization::AnimatedChartContent, 31

- visualization::Chart, 68
- visualization::ChartData, 77
- setBounds
 - visualization::Chart, 68
- setBrown
 - model::BoundaryFileParameter, 49
- setChartTitle
 - visualization::ChartContent, 73
- setColor
 - visualization::ChartData, 77
- setContainer
 - visualization::ChartContent, 73
 - visualization::ChartData, 78
- setContent
 - visualization::AnimatedChart, 28
 - visualization::AnimatedScene, 33
 - visualization::Animation, 39
 - visualization::Chart, 68
- setCurrentPosition
 - visualization::AnimatedChart, 28
 - visualization::AnimatedScene, 33
 - visualization::Animation, 39
- setDefaultConfiguration
 - io::Procedures, 252
- setEnabled
 - model::Dependency, 89
 - model::ExternalNode, 102
- setFile
 - model::BoundaryFileParameter, 50
- setFormula
 - model::FormulaFileBuilderParameter, 132
- setImagesPerSecond
 - visualization::Animation, 39
- setLastDirectoryUsed
 - io::Procedures, 252
- setLaunchingCommands
 - io::Procedures, 252
- setModel
 - io::Procedures, 252
- setName
 - visualization::ChartData, 78
- setOutputDirectory
 - io::Procedures, 253
- setPath
 - model::BoundaryParameter, 57
- setPathRedirection
 - io::Procedures, 253
- setPathRedirectionBrowse
 - io::Procedures, 253
- setPreferences
 - io::Procedures, 253
- setPreferredSize
 - visualization::Chart, 69
- setRain
 - model::RainFileParameter, 261
- setRecentFilesList
 - io::Procedures, 254
- setSimultaneousVisualization
 - io::FullswofIO, 178
- setSize
 - visualization::Chart, 69
- setTime
 - model::BoundaryFileParameter, 50
 - model::RainFileParameter, 261
- setTitle
 - visualization::Chart, 69
- setUpController
 - model::BoundaryFileParameter, 50
 - model::BoundaryParameter, 57
 - model::DirectoryExtensionParameter, 95
 - model::FieldParameter, 108
 - model::FileParameter, 120
 - model::FloatParameter, 126
 - model::FormulaFileBuilderParameter, 132
 - model::IntegerParameter, 194
 - model::InternalNode, 198
 - model::MultipleChoiceParameter, 211
 - model::Node, 216
 - model::PointFileParameter, 238
 - model::RainFileParameter, 262
 - model::RootNode, 271
- setUpView
 - ui::BoundaryFileController, 45
 - ui::FieldParameterController, 109
 - ui::FileParameterController, 123
 - ui::FormulaFileBuilderController, 128
 - ui::MultipleChoiceParameterController, 213
 - ui::NodeController, 219
 - ui::ParametersGroupController, 232
 - ui::PointFileController, 234
 - ui::RainFileController, 258
 - ui::RootController, 269
 - visualization::VisualizationFile, 297
 - visualization::fs1d::FS1DFile, 174
 - visualization::fs2d::FS2DFile, 176
- setValue
 - model::ExternalNode, 102
 - model::FileBuilderParameter, 115
 - model::MultipleChoiceParameter, 211
- setValueAt
 - ui::BoundaryFileController::BoundaryModel, 55
 - ui::PointFileController::PointModel, 241
 - ui::RainFileController::RainModel, 267
- setVerboseParametersFiles
 - io::Procedures, 254
- setVisualizer

- model::FileParameter, 120
- setT
 - visualization::TimeStep, 283
- SettingDependency
 - model::SettingDependency, 272
- SettingDependency2
 - model::SettingDependency2, 274
- showDialog
 - io::DataSetBuilderDialog, 84
- showFr
 - visualization::VisualizationPane, 303
- showHZ
 - visualization::VisualizationPane, 303
- showPopupMenu
 - visualization::Chart, 70
- showQx
 - visualization::VisualizationPane, 304
- showQy
 - visualization::VisualizationPane, 304
- showH
 - visualization::VisualizationPane, 303
- showN
 - visualization::VisualizationPane, 303
- showQ
 - visualization::VisualizationPane, 304
- showU
 - visualization::VisualizationPane, 304
- showV
 - visualization::VisualizationPane, 304
- showZ
 - visualization::VisualizationPane, 304
- signedAtom
 - parser::FormulaParser, 155
- simultaneousVisualization
 - io::FullswofIO, 179
- slave
 - model::Dependency, 90
- slaveValue
 - model::SettingDependency, 273
- src/io/DataSetBuilderDialog.java, 307
- src/io/FullswofIO.java, 307
- src/io/HtmlAbout.java, 308
- src/io/HtmlFrame.java, 309
- src/io/MainFrame.java, 310
- src/io/PreferencesDialog.java, 316
- src/io/Procedures.java, 317
- src/io/ProgressDialog.java, 318
- src/io/Start.java, 319
- src/io/VisualizationFrame.java, 319
- src/io/package-info.java, 310
- src/model/BoundaryFileParameter.java, 320
- src/model/BoundaryParameter.java, 321
- src/model/Dependency.java, 323
- src/model/DirectoryExtensionParameter.java, 324
- src/model/DisablingDependency.java, 325
- src/model/EnablingDependency.java, 325
- src/model/ExternalNode.java, 326
- src/model/FieldParameter.java, 327
- src/model/FileBuilderParameter.java, 327
- src/model/FileParameter.java, 328
- src/model/FloatParameter.java, 329
- src/model/FormulaFileBuilderParameter.java, 330
- src/model/HU1DBuilderParameter.java, 331
- src/model/HUV2DBuilderParameter.java, 331
- src/model/IntegerParameter.java, 332
- src/model/InternalNode.java, 333
- src/model/Interval.java, 333
- src/model/ListFile.java, 334
- src/model/MultipleChoiceParameter.java, 335
- src/model/Node.java, 336
- src/model/PointFileParameter.java, 336
- src/model/RainFileParameter.java, 337
- src/model/RootNode.java, 338
- src/model/SettingDependency.java, 339
- src/model/SettingDependency2.java, 339
- src/model/Topography1DBuilderParameter.java, 340
- src/model/Topography2DBuilderParameter.java, 341
- src/model/definition/Definition_1D.java, 322
- src/model/definition/Definition_2D.java, 322
- src/model/definition/package-info.java, 311
- src/model/package-info.java, 312
- src/parser/FormulaLexer.java, 342
- src/parser/FormulaParser.java, 342
- src/parser/package-info.java, 313
- src/ui/BoundaryFileController.java, 343
- src/ui/BoundaryFileParameterController.java, 344
- src/ui/DirectoryExtensionController.java, 344
- src/ui/ExternalNodeController.java, 345
- src/ui/FieldParameterController.java, 346
- src/ui/FileBuilderController.java, 347
- src/ui/FileBuilderWithoutSavedController.java, 347
- src/ui/FileParameterController.java, 348
- src/ui/FormulaFileBuilderController.java, 349
- src/ui/InternalNodeController.java, 349
- src/ui/MultipleChoiceParameterController.java, 350
- src/ui/NodeController.java, 351
- src/ui/ParametersGroupController.java, 352
- src/ui/PointFileController.java, 352
- src/ui/RainFileController.java, 353
- src/ui/RootController.java, 354
- src/ui/package-info.java, 314
- src/visualization/AnimatedChart.java, 355
- src/visualization/AnimatedChartContent.java, 355
- src/visualization/AnimatedScene.java, 356
- src/visualization/AnimatedSceneContent.java, 357
- src/visualization/Animation.java, 357

- src/visualization/AnimationContent.java, 358
- src/visualization/Chart.java, 359
- src/visualization/ChartContent.java, 360
- src/visualization/ChartData.java, 360
- src/visualization/ChartLine.java, 361
- src/visualization/ChartScatterData.java, 362
- src/visualization/DataFileReader.java, 363
- src/visualization/GnuplotFileReader.java, 376
- src/visualization/InputFileVisualizer.java, 377
- src/visualization/JRealityViewingComponent.java, 378
- src/visualization/OutputFileReader.java, 379
- src/visualization/OutputPoint.java, 379
- src/visualization/TimeLine.java, 380
- src/visualization/TimeStep.java, 381
- src/visualization/VisualizationFile.java, 382
- src/visualization/VisualizationPane.java, 382
- src/visualization/fs1d/FS1DFile.java, 363
- src/visualization/fs1d/FS1DVisualizationPane.java, 364
- src/visualization/fs1d/GnuplotFileReader1D.java, 365
- src/visualization/fs1d/HU1DVisualizer.java, 365
- src/visualization/fs1d/OutputPoint1D.java, 366
- src/visualization/fs1d/RainFileVisualizer.java, 367
- src/visualization/fs1d/TimeLine1D.java, 368
- src/visualization/fs1d/TimeStep1D.java, 369
- src/visualization/fs1d/Topography1DVisualizer.java, 369
- src/visualization/fs1d/package-info.java, 315
- src/visualization/fs2d/FS2DFile.java, 370
- src/visualization/fs2d/FS2DVisualizationPane.java, 371
- src/visualization/fs2d/GnuplotFileReader2D.java, 371
- src/visualization/fs2d/HUV2DVisualizer.java, 372
- src/visualization/fs2d/OutputPoint2D.java, 373
- src/visualization/fs2d/RainFileVisualizer.java, 367
- src/visualization/fs2d/TimeLine2D.java, 374
- src/visualization/fs2d/TimeStep2D.java, 374
- src/visualization/fs2d/Topography2DVisualizer.java, 375
- src/visualization/fs2d/VtkFileReader2D.java, 376
- src/visualization/fs2d/package-info.java, 315
- src/visualization/package-info.java, 316
- startWatching
 - visualization::GnuplotFileReader, 181
 - visualization::OutputFileReader, 222
 - visualization::VisualizationFile, 297
 - visualization::fs2d::VtkFileReader2D, 306
- stateChanged
 - io::MainFrame, 207
 - model::Dependency, 89
 - ui::ExternalNodeController, 106
 - visualization::VisualizationFile, 297
- stopWatching
- visualization::GnuplotFileReader, 182
- visualization::OutputFileReader, 222
- visualization::VisualizationFile, 298
- visualization::fs2d::VtkFileReader2D, 306
- T__10
 - parser::FormulaLexer, 145
 - parser::FormulaParser, 169
- T__11
 - parser::FormulaLexer, 145
 - parser::FormulaParser, 169
- T__12
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__13
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__14
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__15
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__16
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__17
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__18
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 169
- T__19
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 170
- T__20
 - parser::FormulaLexer, 146
 - parser::FormulaParser, 170
- T__21
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__22
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__23
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__24
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__25
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__26

- parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__27
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 170
- T__28
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 171
- T__29
 - parser::FormulaLexer, 147
 - parser::FormulaParser, 171
- T__30
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__31
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__32
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__33
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__34
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__35
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__36
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 171
- T__37
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 172
- T__38
 - parser::FormulaLexer, 148
 - parser::FormulaParser, 172
- T__39
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172
- T__40
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172
- T__41
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172
- T__42
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172
- T__8
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172

- T__9
 - parser::FormulaLexer, 149
 - parser::FormulaParser, 172
- tag
 - model::ExternalNode, 104
- targetValue
 - model::Dependency, 90
- textField
 - visualization::Animation, 41
- time
 - model::RainFileParameter, 262
 - ui::BoundaryFileController::BoundaryModel, 55
 - ui::RainFileController::RainModel, 268
- TimeLine
 - visualization::TimeLine, 276
- TimeLine1D
 - visualization::fs1d::TimeLine1D, 277
- TimeStep
 - visualization::TimeStep, 283
- timeline
 - visualization::GnuplotFileReader, 182
 - visualization::OutputFileReader, 223
 - visualization::VisualizationFile, 298
- toFile
 - model::ExternalNode, 103
 - model::FileBuilderParameter, 115
 - model::FileParameter, 120
 - model::InternalNode, 198
 - model::MultipleChoiceParameter, 211
 - model::Node, 216
- toString
 - model::DisablingDependency, 96
 - model::EnablingDependency, 98
 - model::MultipleChoiceParameter::PossibleValue, 244
 - model::Node, 216
- tokenNames
 - parser::FormulaParser, 172
- Topography1DBuilderParameter
 - model::Topography1DBuilderParameter, 291
- Topography2DBuilderParameter
 - model::Topography2DBuilderParameter, 294
- u
 - visualization::fs2d::OutputPoint2D, 231
- uColor
 - visualization::VisualizationPane, 304
- ui, 22
 - ui.BoundaryFileController, 43
 - ui.BoundaryFileController.BoundaryModel, 52
 - ui.BoundaryFileController.CellRenderer, 60
 - ui.BoundaryFileParameterController, 50
 - ui.DirectoryExtensionController, 92
 - ui.ExternalNodeController, 105

- ui.FieldParameterController, 108
- ui.FileBuilderController, 110
- ui.FileBuilderWithoutSavedController, 116
- ui.FileParameterController, 121
- ui.FormulaFileBuilderController, 127
- ui.InternalNodeController, 199
- ui.MultipleChoiceParameterController, 212
- ui.NodeController, 217
- ui.ParametersGroupController, 231
- ui.PointFileController, 233
- ui.PointFileController.CellRenderer, 59
- ui.PointFileController.PointModel, 239
- ui.RainFileController, 256
- ui.RainFileController.CellRenderer, 58
- ui.RainFileController.RainModel, 264
- ui.RootController, 268
- ui::BoundaryFileController
 - BoundaryFileController, 44
 - highlightView, 44
 - mouseClicked, 44
 - mouseEntered, 45
 - mouseExited, 45
 - mousePressed, 45
 - mouseReleased, 45
 - setUpView, 45
 - updateModel, 45
 - updateView, 45
 - validate, 45
 - viewLabel, 46
 - viewTable, 46
- ui::BoundaryFileController::BoundaryModel
 - BoundaryModel, 53
 - file, 55
 - getColumnClass, 53
 - getColumnCount, 53
 - getColumnName, 53
 - getRowCount, 53
 - getValueAt, 54
 - isCellEditable, 54
 - isValidCell, 54
 - setValueAt, 55
 - time, 55
- ui::BoundaryFileController::CellRenderer
 - getTableCellRendererComponent, 61
- ui::BoundaryFileParameterController
 - BoundaryFileParameterController, 51
 - validate, 51
- ui::DirectoryExtensionController
 - DirectoryExtensionController, 92
 - validate, 93
- ui::ExternalNodeController
 - ExternalNodeController, 105
 - focusLost, 105
 - highlightView, 106
 - stateChanged, 106
 - updateModel, 106
 - updateView, 106
 - validate, 106
- ui::FieldParameterController
 - FieldParameterController, 109
 - highlightView, 109
 - setUpView, 109
 - updateModel, 109
 - updateView, 109
 - viewField, 110
 - viewLabel, 110
- ui::FileBuilderController
 - FileBuilderController, 111
 - validate, 111
- ui::FileBuilderWithoutSavedController
 - FileBuilderWithoutSavedController, 116
 - validate, 116
- ui::FileParameterController
 - actionPerformed, 122
 - copyFile, 122
 - FileParameterController, 122
 - messages, 123
 - setUpView, 123
 - updateModel, 123
 - updateView, 123
 - validate, 123
- ui::FormulaFileBuilderController
 - actionPerformed, 128
 - FormulaFileBuilderController, 128
 - highlightView, 128
 - setUpView, 128
 - updateModel, 128
 - updateView, 128
- ui::InternalNodeController
 - addController, 200
 - controllerList, 201
 - InternalNodeController, 200
 - validate, 200
- ui::MultipleChoiceParameterController
 - actionPerformed, 213
 - highlightView, 213
 - MultipleChoiceParameterController, 212
 - setUpView, 213
 - updateModel, 213
 - updateView, 213
- ui::NodeController
 - focusGained, 218
 - focusLost, 219
 - getModel, 219
 - getView, 219
 - LIGHT_RED, 220

- MIN_BOX_WIDTH, 220
- MIN_FIELD_WIDTH, 220
- MIN_LABEL_WIDTH, 220
- model, 220
- NodeController, 218
- resizeViewComponent, 219
- setUpView, 219
- validate, 220
- view, 221
- ui::ParametersGroupController
 - ParametersGroupController, 232
 - setUpView, 232
- ui::PointFileController
 - highlightView, 234
 - PointFileController, 234
 - setUpView, 234
 - updateModel, 234
 - updateView, 234
 - viewLabel, 234
 - viewTable, 234
- ui::PointFileController::CellRenderer
 - getTableCellRendererComponent, 60
- ui::PointFileController::PointModel
 - getColumnClass, 239
 - getColumnCount, 240
 - getColumnName, 240
 - getRowCount, 240
 - getValueAt, 240
 - isCellEditable, 241
 - isValidCell, 241
 - listX, 242
 - listY, 242
 - PointModel, 239
 - setValueAt, 241
- ui::RainFileController
 - highlightView, 257
 - RainFileController, 257
 - setUpView, 258
 - updateModel, 258
 - updateView, 258
 - viewLabel, 258
 - viewTable, 258
- ui::RainFileController::CellRenderer
 - getTableCellRendererComponent, 58
- ui::RainFileController::RainModel
 - getColumnClass, 265
 - getColumnCount, 266
 - getColumnName, 266
 - getRowCount, 266
 - getValueAt, 266
 - isCellEditable, 267
 - isValidCell, 267
 - rain, 268
 - RainModel, 265
 - setValueAt, 267
 - time, 268
- ui::RootController
 - RootController, 269
 - setUpView, 269
 - validate, 269
- update
 - visualization::VisualizationPane, 302
 - visualization::fs1d::FS1DVisualizationPane, 175
 - visualization::fs2d::FS2DVisualizationPane, 178
- updateChangeEvent
 - model::ExternalNode, 103
 - model::InternalNode, 198
 - model::Node, 216
 - model::definition::Definition_2D, 86
- updateContent
 - io::MainFrame, 207
- updateModel
 - ui::BoundaryFileController, 45
 - ui::ExternalNodeController, 106
 - ui::FieldParameterController, 109
 - ui::FileParameterController, 123
 - ui::FormulaFileBuilderController, 128
 - ui::MultipleChoiceParameterController, 213
 - ui::PointFileController, 234
 - ui::RainFileController, 258
- updateProgress
 - io::ProgressDialog, 256
- updateRecentFilesMenu
 - io::MainFrame, 207
- updateTimeline
 - visualization::GnuplotFileReader, 182
 - visualization::OutputFileReader, 222
 - visualization::fs2d::VtkFileReader2D, 306
- updateView
 - ui::BoundaryFileController, 45
 - ui::ExternalNodeController, 106
 - ui::FieldParameterController, 109
 - ui::FileParameterController, 123
 - ui::FormulaFileBuilderController, 128
 - ui::MultipleChoiceParameterController, 213
 - ui::PointFileController, 234
 - ui::RainFileController, 258
 - visualization::VisualizationFile, 298
- v
 - visualization::fs2d::OutputPoint2D, 231
- vColor
 - visualization::VisualizationPane, 304
- VTK
 - visualization::fs2d::FS2DFile::Format, 127
- validate
 - ui::BoundaryFileController, 45

- ui::BoundaryFileParameterController, 51
- ui::DirectoryExtensionController, 93
- ui::ExternalNodeController, 106
- ui::FileBuilderController, 111
- ui::FileBuilderWithoutSavedController, 116
- ui::FileParameterController, 123
- ui::InternalNodeController, 200
- ui::NodeController, 220
- ui::RootController, 269
- value
 - model::ExternalNode, 104
- view
 - ui::NodeController, 221
 - visualization::VisualizationFile, 298
- viewField
 - ui::FieldParameterController, 110
- viewLabel
 - ui::BoundaryFileController, 46
 - ui::FieldParameterController, 110
 - ui::PointFileController, 234
 - ui::RainFileController, 258
- viewTable
 - ui::BoundaryFileController, 46
 - ui::PointFileController, 234
 - ui::RainFileController, 258
- visualization, 23
- visualization.AnimatedChart, 27
- visualization.AnimatedChartContent, 29
- visualization.AnimatedScene, 32
- visualization.AnimatedSceneContent, 34
- visualization.Animation< E >, 34
- visualization.AnimationContent< E >, 41
- visualization.Chart, 61
- visualization.ChartContent, 70
- visualization.ChartData, 74
- visualization.ChartLine, 79
- visualization.ChartScatterData, 81
- visualization.DataFileReader, 83
- visualization.fs1d, 25
- visualization.fs1d.FS1DFile, 173
- visualization.fs1d.FS1DVisualizationPane, 174
- visualization.fs1d.GnuplotFileReader1D, 182
- visualization.fs1d.HU1DVisualizer, 188
- visualization.fs1d.OutputPoint1D, 224
- visualization.fs1d.RainFileVisualizer, 263
- visualization.fs1d.TimeLine1D, 276
- visualization.fs1d.TimeStep1D, 284
- visualization.fs1d.Topography1DVisualizer, 292
- visualization.fs2d, 25
- visualization.fs2d.FS2DFile, 175
- visualization.fs2d.FS2DFile.Format, 126
- visualization.fs2d.FS2DVisualizationPane, 176
- visualization.fs2d.GnuplotFileReader2D, 184
- visualization.fs2d.HUV2DVisualizer, 190
- visualization.fs2d.OutputPoint2D, 226
- visualization.fs2d.RainFileVisualizer, 263
- visualization.fs2d.TimeLine2D, 279
- visualization.fs2d.TimeStep2D, 287
- visualization.fs2d.Topography2DVisualizer, 294
- visualization.fs2d.VtkFileReader2D, 305
- visualization.GnuplotFileReader< OUTPUT_POINT
extends OutputPoint, TIME_STEP extends
TimeStep< OUTPUT_POINT, TIME_LINE
extends TimeLine< TIME_STEP >, 179
- visualization.InputFileVisualizer, 191
- visualization.JRealityViewingComponent, 204
- visualization.OutputFileReader, 221
- visualization.OutputPoint, 223
- visualization.TimeLine< E extends TimeStep<?>, 276
- visualization.TimeStep< E extends OutputPoint >, 282
- visualization.VisualizationFile< E extends Time↔
Line<?>, 295
- visualization.VisualizationPane, 301
- visualization::AnimatedChart
 - addImage, 28
 - AnimatedChart, 27
 - makeBackgroundComponent, 28
 - setContent, 28
 - setCurrentPosition, 28
- visualization::AnimatedChartContent
 - add, 29
 - AnimatedChartContent, 29
 - getXmax, 31
 - getXmin, 31
 - getYmax, 31
 - getYmin, 31
 - setBoundariesToInclude, 31
- visualization::AnimatedScene
 - addImage, 33
 - AnimatedScene, 32
 - makeBackgroundComponent, 33
 - setContent, 33
 - setCurrentPosition, 33
- visualization::AnimatedSceneContent
 - add, 34
 - AnimatedSceneContent, 34
- visualization::Animation
 - actionPerformed, 36
 - addImage, 37
 - Animation, 36
 - backgroundComponent, 39
 - commandPanel, 37
 - content, 40
 - currentPosition, 40

- first, [40](#)
- getContent, [37](#)
- getCurrentPosition, [37](#)
- getImagesCount, [37](#)
- getImagesPerSecond, [38](#)
- hierarchyChanged, [38](#)
- imagesPerSecond, [40](#)
- isPlaying, [38](#)
- last, [40](#)
- loadIcons, [38](#)
- makeBackgroundComponent, [38](#)
- next, [40](#)
- pause, [40](#)
- play, [40](#)
- playOrPause, [39](#)
- playOrPauseButton, [41](#)
- playerThread, [41](#)
- playing, [41](#)
- previous, [41](#)
- setContent, [39](#)
- setCurrentPosition, [39](#)
- setImagesPerSecond, [39](#)
- textField, [41](#)
- visualization::AnimationContent
 - add, [42](#)
 - AnimationContent, [42](#)
 - get, [42](#)
 - getImagesCount, [42](#)
 - removeImages, [43](#)
- visualization::Chart
 - actionPerformed, [63](#)
 - add, [63](#)
 - Chart, [62](#)
 - clear, [63](#)
 - getContent, [63](#)
 - getImage, [64](#)
 - getPreferredSize, [64](#)
 - getSize, [64](#)
 - getTitle, [64](#)
 - getXmax, [64](#)
 - getXmin, [64](#)
 - getYmax, [65](#)
 - getYmin, [65](#)
 - mouseClicked, [65](#)
 - mouseEntered, [65](#)
 - mouseExited, [65](#)
 - mousePressed, [65](#)
 - mouseReleased, [66](#)
 - paint, [66](#)
 - savelmage, [66](#)
 - scaledLine, [66](#)
 - scaledPoint, [67](#)
 - setBoundaries, [67](#)
 - setBoundariesToInclude, [68](#)
 - setBounds, [68](#)
 - setContent, [68](#)
 - setPreferredSize, [69](#)
 - setSize, [69](#)
 - setTitle, [69](#)
 - showPopupMenu, [70](#)
- visualization::ChartContent
 - add, [71](#)
 - ChartContent, [71](#)
 - clear, [71](#)
 - getChartTitle, [71](#)
 - getContainer, [71](#)
 - getData, [72](#)
 - getXmax, [72](#)
 - getXmin, [72](#)
 - getYmax, [72](#)
 - getYmin, [72](#)
 - paint, [72](#)
 - remove, [73](#)
 - setChartTitle, [73](#)
 - setContainer, [73](#)
- visualization::ChartData
 - add, [75](#)
 - ChartData, [74](#), [75](#)
 - color, [78](#)
 - container, [78](#)
 - data, [78](#)
 - getColor, [75](#)
 - getContainer, [76](#)
 - getName, [76](#)
 - getXmax, [76](#)
 - getXmin, [76](#)
 - getYmax, [76](#)
 - getYmin, [76](#)
 - name, [79](#)
 - paint, [77](#)
 - paintLegendElement, [77](#)
 - setBoundariesToInclude, [77](#)
 - setColor, [77](#)
 - setContainer, [78](#)
 - setName, [78](#)
- visualization::ChartLine
 - ChartLine, [79](#), [80](#)
 - paint, [80](#)
 - paintLegendElement, [80](#)
- visualization::ChartScatterData
 - ChartScatterData, [81](#), [82](#)
 - paint, [82](#)
 - paintLegendElement, [82](#)
- visualization::DataFileReader
 - buildDataSet, [83](#)
- visualization::GnuplotFileReader

- GnuplotFileReader, 180
- makePoint, 180
- makeTimeStep, 180
- readTimeStep, 181
- startWatching, 181
- stopWatching, 182
- timeline, 182
- updateTimeline, 182
- visualization::InputFileVisualizer
 - getVisualization, 192
- visualization::JRealityViewingComponent
 - makeViewingComponent, 204
- visualization::OutputFileReader
 - addChangeListener, 222
 - changeEvent, 223
 - fireChangeEvent, 222
 - listenerList, 223
 - removeChangeListener, 222
 - startWatching, 222
 - stopWatching, 222
 - timeline, 223
 - updateTimeline, 222
- visualization::TimeLine
 - TimeLine, 276
- visualization::TimeStep
 - getT, 283
 - setT, 283
 - TimeStep, 283
- visualization::VisualizationFile
 - getName, 296
 - getTimeline, 296
 - getView, 296
 - name, 298
 - open, 297
 - reader, 298
 - setUpView, 297
 - startWatching, 297
 - stateChanged, 297
 - stopWatching, 298
 - timeline, 298
 - updateView, 298
 - view, 298
- visualization::VisualizationPane
 - defaultLineColor, 302
 - frColor, 302
 - hColor, 302
 - hzColor, 302
 - nColor, 303
 - qColor, 303
 - qxColor, 303
 - qyColor, 303
 - showFr, 303
 - showHZ, 303
 - showQx, 304
 - showQy, 304
 - showH, 303
 - showN, 303
 - showQ, 304
 - showU, 304
 - showV, 304
 - showZ, 304
 - uColor, 304
 - update, 302
 - vColor, 304
 - zColor, 305
- visualization::fs1d::FS1DFile
 - FS1DFile, 173
 - setUpView, 174
- visualization::fs1d::FS1DVisualizationPane
 - actionPerformed, 175
 - FS1DVisualizationPane, 174
 - update, 175
- visualization::fs1d::GnuplotFileReader1D
 - GnuplotFileReader1D, 183
 - makePoint, 183
 - makeTimeStep, 184
- visualization::fs1d::HU1DVisualizer
 - getVisualization, 189
- visualization::fs1d::OutputPoint1D
 - compareTo, 225
 - equals, 225
 - getDx, 225
 - getFr, 225
 - getHZ, 225
 - getH, 225
 - getQ, 226
 - getU, 226
 - getZ, 226
 - OutputPoint1D, 224
- visualization::fs1d::RainFileVisualizer
 - getVisualization, 263
- visualization::fs1d::TimeLine1D
 - add, 277
 - getVisualisation, 277
 - getXmaxCumulativeDischarge, 278
 - getXmaxInstantDischarge, 278
 - getXmaxWaterHeight, 278
 - getXminCumulativeDischarge, 278
 - getXminInstantDischarge, 278
 - getXminWaterHeight, 279
 - TimeLine1D, 277
- visualization::fs1d::TimeStep1D
 - getChartLineFr, 284
 - getChartLineHZ, 285
 - getChartLineH, 284
 - getChartLineQ, 285

- getChartLineU, [285](#)
- getChartLineZ, [285](#)
- getVisualization, [285](#)
- getXmaxDischarge, [286](#)
- getXmaxWaterHeight, [286](#)
- getXminDischarge, [286](#)
- getXminWaterHeight, [286](#)
- visualization::fs1d::Topography1DVisualizer
 - getVisualization, [292](#)
- visualization::fs2d::FS2DFile
 - FS2DFile, [176](#)
 - setUpView, [176](#)
- visualization::fs2d::FS2DFile::Format
 - GNUPLOT, [127](#)
 - VTK, [127](#)
- visualization::fs2d::FS2DVisualizationPane
 - actionPerformed, [177](#)
 - FS2DVisualizationPane, [177](#)
 - update, [178](#)
- visualization::fs2d::GnuplotFileReader2D
 - GnuplotFileReader2D, [185](#)
 - makePoint, [185](#)
 - makeTimeStep, [185](#)
- visualization::fs2d::HUV2DVisualizer
 - getVisualization, [191](#)
- visualization::fs2d::OutputPoint2D
 - compareTo, [228](#)
 - dx, [230](#)
 - dy, [231](#)
 - equals, [228](#)
 - getDx, [228](#)
 - getDy, [228](#)
 - getFr, [229](#)
 - getHZ, [229](#)
 - getQx, [229](#)
 - getQy, [230](#)
 - getH, [229](#)
 - getN, [229](#)
 - getQ, [229](#)
 - getU, [230](#)
 - getV, [230](#)
 - getZ, [230](#)
 - h, [231](#)
 - OutputPoint2D, [227](#)
 - u, [231](#)
 - v, [231](#)
 - z, [231](#)
- visualization::fs2d::RainFileVisualizer
 - getVisualization, [264](#)
- visualization::fs2d::TimeLine2D
 - getVisualisation, [280](#)
 - getXmaxCumulativeDischarge, [280](#)
 - getXmaxInstantDischarge, [280](#)
 - getXmaxWaterHeight, [280](#)
 - getXminCumulativeDischarge, [280](#)
 - getXminInstantDischarge, [281](#)
 - getXminWaterHeight, [281](#)
 - getYmaxCumulativeDischarge, [281](#)
 - getYmaxInstantDischarge, [281](#)
 - getYmaxWaterHeight, [281](#)
 - getYminCumulativeDischarge, [282](#)
 - getYminInstantDischarge, [282](#)
 - getYminWaterHeight, [282](#)
- visualization::fs2d::TimeStep2D
 - getFrSceneGraphComponent, [287](#)
 - getHSceneGraphComponent, [287](#)
 - getHZSceneGraphComponent, [288](#)
 - getNSceneGraphComponent, [288](#)
 - getQSceneGraphComponent, [288](#)
 - getQxSceneGraphComponent, [288](#)
 - getQySceneGraphComponent, [288](#)
 - getUSceneGraphComponent, [289](#)
 - getVSceneGraphComponent, [289](#)
 - getVisualization, [289](#)
 - getXmaxDischarge, [289](#)
 - getXmaxWaterHeight, [289](#)
 - getXminDischarge, [290](#)
 - getXminWaterHeight, [290](#)
 - getYmaxDischarge, [290](#)
 - getYmaxWaterHeight, [290](#)
 - getYminDischarge, [290](#)
 - getYminWaterHeight, [290](#)
 - getZSceneGraphComponent, [291](#)
- visualization::fs2d::Topography2DVisualizer
 - getVisualization, [295](#)
- visualization::fs2d::VtkFileReader2D
 - startWatching, [306](#)
 - stopWatching, [306](#)
 - updateTimeline, [306](#)
 - VtkFileReader2D, [305](#)
- VisualizationFrame
 - io::VisualizationFrame, [299](#)
- visualizer
 - model::FileParameter, [121](#)
 - model::FormulaFileBuilderParameter, [132](#)
- VtkFileReader2D
 - visualization::fs2d::VtkFileReader2D, [305](#)
- windowActivated
 - io::MainFrame, [207](#)
 - io::VisualizationFrame, [299](#)
- windowClosed
 - io::MainFrame, [207](#)
 - io::VisualizationFrame, [299](#)
- windowClosing
 - io::MainFrame, [208](#)
 - io::VisualizationFrame, [300](#)

- windowDeactivated
 - io::MainFrame, [208](#)
 - io::VisualizationFrame, [300](#)
- windowDeiconified
 - io::MainFrame, [208](#)
 - io::VisualizationFrame, [300](#)
- windowIconified
 - io::MainFrame, [208](#)
 - io::VisualizationFrame, [300](#)
- windowOpened
 - io::MainFrame, [208](#)
 - io::VisualizationFrame, [300](#)
- workingModel
 - io::Procedures, [254](#)
- z
 - visualization::fs2d::OutputPoint2D, [231](#)
- zColor
 - visualization::VisualizationPane, [305](#)