**System Dynamics Java Source File Information**

* Location of directory for source files: EASimulator\Development\2013-12-09\  
  SD-S10\src\de\uka\aifb\com\systemDynamics
* Yellow highlights indicate particularly important files/files that have 1000 lines or more of code
* Contents of the directory of source files
  + csv (directory)
    - CSVExport.java
      * Exports to CSV. Numbers of type Double are formatted with 9 places after the decimal point. CSV file is semicolon delimited.
  + event (directory)
    - ExportModelExecutionThreadEventListener.java
      * Describes an event listener that listens for export model execution events. Just contains method and interface declarations.
    - SystemDynamicsGraphModifiedListener.java
      * describes an event listener that listens for modified graph events. Just contains method and interface declarations.
  + gui (directory)
    - SystenDynamicsGraph (directory)
      * AutomaticGraphLayout.java:
        + implements automatic graph layout using algorithm described in the following paper:

FRUCHTERMAN, Thomas M. J. ; REINGOLD, Edward M.: "Graph Drawing by Force-directed Placement". In: Software - Practice and Experience, vol. 21, no. 11, November 1991, pp. 1129-1164.

* + - * AuxiliaryNodeGraphCell.java
        + implements graph cell for auxiliary node
      * AuxiliaryNodeVertexRenderer.java
        + implements specialized vertex renderer for auxiliary node.
      * AuxiliaryNodeVertexView.java
        + implements a specialized vertex view for an auxiliary node.
      * CloudShapeFactory.java
        + code for generating the cloud shape
      * ConstantNodeGraphCell.java
        + implements graph cell for constant node
      * ConstantNodeVertexView.java
        + implements vertex view for constant node
      * FlowEdge.java
        + implements specialized edge for flows
      * LevelNodeGraphCell.java
        + implements graph cell for a level node
      * MyTableModel.java:
        + Implements model for non-editable table. Adds isCellEditable method to DefaultTableModel
      * NodeFormulaDialog.java:
        + Dialog for entering new node formula
      * NodeNameDialog.java
        + Dialog for entering new node name
      * NodeParameterDialog.java
        + Dialog for entering new node parameters
      * RateNodeGraphCell.java
        + Graph cell specialized for rate node
      * RateNodeVertexRenderer.java
        + Vertex renderer for a rate node.
      * RateNodeVertexView.java
        + Vertex view for a rate node.
      * ShiftEdgeHandle.java
        + Shift + mouseclick adds or removes additional control points. This file implements edge handles.
      * SourceSinkNodeGraphCell.java
        + Graph cell for source/sink node
      * SourceSinkNodeVertexRenderer.java
        + Implements vertex renderer for source/sink node
      * SourceSinkNodeVertexView
        + Implements vertex view for source/sink node.
      * SystemDynamicsCellViewFactory.java
        + Returns different cell views for each vertex type.
      * SystemDynamicsEdgeView.java
        + implements edge view. Shift+click adds/removes additional control points of an edge.
      * SystemDynamicsGraph.java
        + implements SystemDynamics version of the JGraph api.
      * SystemDynamicsMarqueHandler.java
        + Implements SystemDynamics version of marquee handler.
    - CSVFileFilter.java
      * implements file filter for CSV files
    - ExportPanel.java
      * Implements panel for exporting the values of a model execution (also actually executes the model for x rounds).
    - MainFrame.java:
      * Main frame of the application
    - ModelExecutionChartPanel.java
      * implements a panel for drawing the charts of the model execution
    - ModelNameDialog.java
      * Dialog for entering new model name
    - NodeNameDialog.java
      * Dialog for entering a node name for a new node
    - NodeNameParameterDialog.java
      * Dialog for entering node name and a parameter for a new node.
    - XMLFileFilter.java
      * file filter for XML files
  + model (directory)
    - *Note: the AST in some file names stands for Abstract Syntax Tree. For more information on how an abstract syntax tree works, see* [*http://en.wikipedia.org/wiki/Abstract\_syntax\_tree*](http://en.wikipedia.org/wiki/Abstract_syntax_tree)*.*
    - AbstractNode.java
      * implements an abstract node of a System Dynamics model
    - ASTElement.java
      * Describes an abstract syntax tree element
    - ASTMinus.java
      * AST element representing subtraction
    - ASTMultiply.java
      * AST element representing multiplication
    - ASTPlus.java
      * AST element representing addition
    - AuxiliaryNode.java
      * implements model representing auxiliary node
    - AuxiliaryNodesCycleDependancyException.java
      * Error message that is thrown if there is a cycle that exists among a group of auxiliary nodes.
    - ConstantNode.java
      * Implements a model node representing a constant
    - FormulaDependencyException.java
      * Error message that says a node cannot be removed from a model because it is part of another node’s formula.
    - LevelNode.java
      * Implements a model node representing a level
    - Model.java
      * Implements a model
    - ModelNotChangeableException.java
      * Indicates model was in unchangeable state when a method tried to change it
    - ModelStillChangeableException.java
      * Indicates that the model has not been set unchangeable before a method tried to use it.
    - nodeParameterOutOfRangeException.java
      * level node’s start value or a constant node’s constant value is out of range.
    - NoFormulaException.java
      * Indicates that rate node or auxiliary node has no formula
    - NoLevelNodeException.java
      * Indicates that model has no level node
    - RateNode.java
      * implements a model node representing a change rate
    - RateNodeFlowException.java
      * Rate node has no incoming or outgoing flow
    - SourceSinkNode.java
      * implements model node representing source/sink
    - UselessNodeException.java
      * Indicates a node of the model is useless (i.e. has no influence on a level node). Only constant and auxiliary nodes can be useless.
  + parser (directory)
    - *Note: all files in this directory have been automatically generated.*
    - FormulaParser.java
      * parser for node formulas
    - FormulaParser.jj
      * same as above, but in a different file format
    - FormulaParserConstants.java
      * Establishes constants used in parsing
    - FormulaParserTokenManager.java
      * Establishes tokens for different elements of the formula to be parsed
    - ParseException.java
      * Indicates that the formula is incorrectly formatted.
    - SimpleCharStream.java
      * Implementation of CharStream where the stream only contains ASCII characters.
    - Token.java
      * Describes input token stream
    - TokenMgrError.java
      * Token Manager error
  + test (directory)
    - *Note: any java files that don’t have descriptions are files that test the accuracy of specific classes. The class that each file tests is listed in its specific filename.*
    - mocks (directory)
      * RateNodeMockObject.java: mock object for a rate node that always returns a specified “current” value
    - ASTMinusTestCase.java
    - ASTMultiplyTestCase.java
    - ASTPlusTestCase.java
    - AuxiliaryNodeTestCase.java
    - ConstantNodeTestCase.java
    - CSVExportTestCase.java
    - FormulaParserTestCase.java
    - LevelNodeTestCase.java
    - ModelTestCase.java
    - NodeParameterOutOfRangeExceptionTestCase.java
    - RateNodeTestCase.java
    - RunAllTestCases.java
      * executes all existing test cases
    - SourceSinkNodeTestCase.java
    - XMLExportTestCase.java
    - XMLModelReaderTestCase.java
    - XMLModelReaderWriterExceptionTestCase.java
    - XMLWriterTestCase.java
    - XMLNodeParameterOutOfRangeExceptionTestCase.java
    - XMLRateNodeFlowExceptionTestCase.java
    - XMLUselessNodeExceptionTestCase.java
  + xml (directory)
    - MyErrorHandler.java
      * implements the ErrorHandler interface. “puts through” any SAXParseException for methods error and fatal error. Necessary because Java 5 and Java 6 have different error handlers that cause different behavior in non-schema compliant XML files
    - XMLExport.java: implements XML format export to a file. System’s standard encoding and line separator are used
    - XMLModelReader.java: XML input for models stored in an XML file.
    - XMLModelReaderWriterException.java
      * Indicates an error occurred during reading a model from an XML file or writing a model into an XML file
    - XMLModelWriter.java: implements an XML output for SystemDynamics models to store it in an XML file
    - XMLNodeParameterOutOfRangeException.java
      * Either a level node’s start value or a constant node’s constant value is out of range.
    - XMLRateNodeFlowException.java
      * rate node has no incoming or outgoing flow
    - XMLUselessNodeException.java
      * Node of a model has no influence on level node
  + SystemDynamics.java
    - Starting point for application. Initializes new mainframe
  + SystemDynamicsCommandLine.java
    - Command line model execution and value export