Bundle analytical tool for Flare API

User GUIDE

Contents

[Introduction 2](#_Toc497751475)

[Glossary 3](#_Toc497751476)

[1. Interaction with the tool 4](#_Toc497751477)

[1.1. UI description 4](#_Toc497751478)

[1.2. Relationship visualization 7](#_Toc497751479)

[2. Packages and classes description 9](#_Toc497751480)

[2.1. Package flare.analytics.cluster 9](#_Toc497751482)

[2.2. Package flare.analytics.optimization 9](#_Toc497751483)

[2.3. Package flare.data 9](#_Toc497751484)

[2.4. Package flare.animate 9](#_Toc497751485)

[2.5. Package flare.display 10](#_Toc497751486)

[2.6. Package flare.query 10](#_Toc497751487)

[2.7. Package flare.scale 11](#_Toc497751488)

[2.8. Package flare.util 11](#_Toc497751489)

[2.9. Package flare.util.palette 12](#_Toc497751490)

[2.10. Package flare.vis 12](#_Toc497751491)

[2.11. Package flare.vis.controls 12](#_Toc497751492)

[2.12. Package flare.vis.legend 13](#_Toc497751493)

[2.13. Package flare.vis.operator.label 13](#_Toc497751494)

[2.14. Package flare.vis.operator.layout 13](#_Toc497751495)

# Introduction

**Flare** is an ActionScript library for creating visualizations that run in the Adobe Flash Player. From basic charts and graphs to complex interactive graphics, the toolkit supports data management, visual encoding, animation, and interaction techniques.

Flare API contains various **classes**, that are combined in **packages**, and may be used for creating your visualizations. Although each package contains classes, that are created to solve the particular visualization tasks, some classes depend on the others. These dependencies may be complex and unobvious, so a visualization tool was created for better understanding of Flare API classes’ relationships.

In section 1.1 the UI (user interface) of the tool is described.

In section 1.2 relationship visualization is described

In section 2 Flare API packages’ features are covered.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Application programming interface (API) | A set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. |

# Interaction with the tool

## UI description

Tool’s UI is a single web-page with the following elements:

* **A radial layout.**

Classes names are written on the **outer edge** of the layout. All classes are grouped in accordance with their packages. You can distinguish one package from another since they are located at a greater distance than classes within one package (see red boxes in the picture below).

Relations between the classes are visualized **as bundles** **within the circle**.

Also, you can rotate the circle for the better representation of the packages. To do so, move your mouse pointer (the pointer transforms from  to ) in any place within the circle, press the left mouse button and rotate clockwise or counter-clockwise.

* **A tension slider.**

This slider controls the **bundling strength**. Bundling reduces visual clutter, making it easier to perceive the actual connections. Low bundling strength mainly provides low-level, class-to-class connectivity information, whereas high bundling strength provides high-level information as well by implicit visualization of adjacency edges between **parent** classes that are the result of explicit adjacency edges between their respective **child** classes.

It is up to you to decide what bundling strength is necessary for your needs.

The closer the handler to the slider’s left edge, the lower is bundling strength. As the bundling strength is increased, areas of interest emerge as bundles. You can control the bundling strength:

* + by simply pressing the left mouse button and dragging the handler;
  + by arrow buttons on the keyboard. Click anywhere in the slider area and then use the arrows (right/up arrow button to increase the strength and left/down arrow button to decrease it).

Some browsers (i.e., Microsoft Edge v. 40) also visualize the current tension value from 0 to 100.

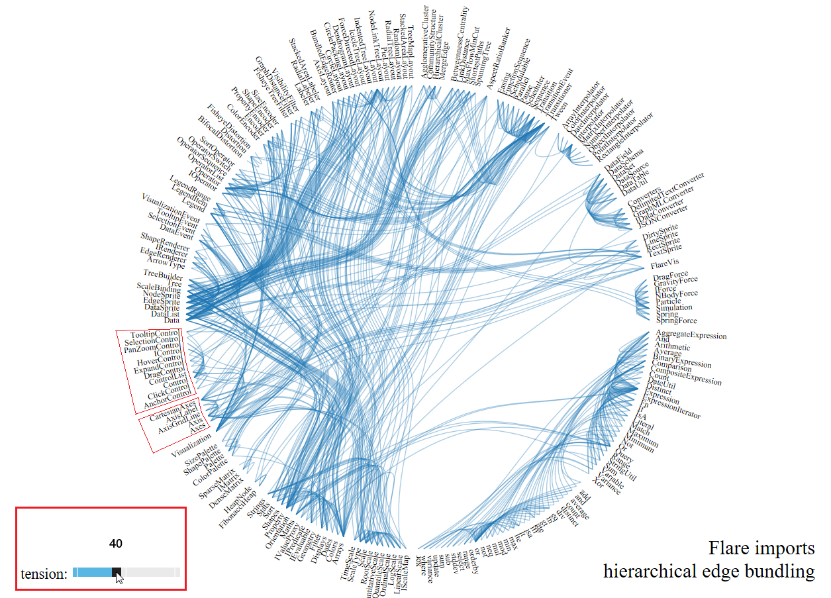


Figure 1. UI elements.

## Relationship visualization

You can visualize connections for a particular class. To see the connection, one should put the mouse pointer on the class. Next:

* The selected class is highlighted blue;
* The dependant classes (or *callers*) are highlighted green;
* The superior classes (or *callees*) are highlighted red.

**Example 1. How to see the callers**.

A «Palette» class with the highest tension is chosen.

Callers **within the package** are SizePalette, ShapePalette, ColorPalette.

Callers **outside the package** are SizeEncoder, ShapeEncoder, Encoder, ColorEncoder.

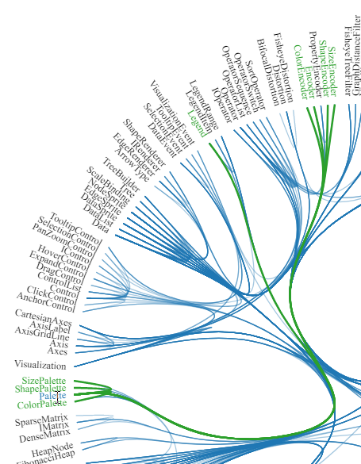


Figure 2. Callers visualization.

**Example 2**. **How to see the callees**.

A «TooltipControl» class with medium tension is chosen.

Callee **within the package** is Control.

Callees **outside the package** are TooltipEvent, Tween, TextSprite.

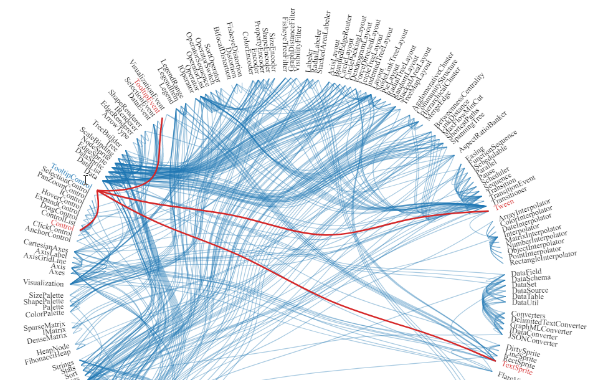


Figure 3. Callees visualization.

# Packages and classes description

In this section packages and classes are described. Please note, that only visualization, animation, and some querying packages are covered here, the «technical» ones are excluded. For more information on packages refer to <http://flare.prefuse.org/api/>.



## Package flare.analytics.cluster

The flare.analytics.cluster package provides classes for clustering data into groups.

|  |  |
| --- | --- |
| **Class** | **Description** |
| AgglomerativeCluster | Hierarchically clusters a set of items using agglomerative clustering. |
| CommunityStructure | Hierarchically clusters a network based on inferred community structure. |
| HierarchicalCluster | Base class for clustering operators that construct a hierarchical clustering tree. |

## Package flare.analytics.optimization

The flare.analytics.optimization package provides classes for optimizing the presentation of visualizations.

|  |  |
| --- | --- |
| **Class** | **Description** |
| AspectRatioBanker | Computes an optimized aspect ratio for drawing a line chart. |

## Package flare.data

The flare.data package provides classes for describing and importing external data sets.

|  |  |
| --- | --- |
| **Class** | **Description** |
| DataField | A data field stores metadata for an individual data variable. |
| DataSchema | A data schema represents a set of data variables and their associated types. |
| DataSet | A data set is a collection of data tables. |
| DataSource | A data source provides access to remote data on the Internet. |
| DataTable | A data table maintains a collection of data objects, each representing a row of data, and an optional data schema describing the data variables. |
| DataUtil | Utility class for parsing and representing data field values. |

## Package flare.animate

The flare.animate package provides classes for creating animated transitions.

|  |  |
| --- | --- |
| **Class** | **Description** |
| Easing | Collection of easing functions to control animation rates. |
| FunctionSequence | Transition that runs sub-transitions in sequential order while also invoking a function before each sub-transition is run. |
| Parallel | Transition that runs multiple transitions simultaneously (in parallel). |
| Pause | Transition is representing a pause or dwell in which nothing happens. |
| Scheduler | Scheduler that oversees animation and time-based processing. |
| Sequence | Transition that runs multiple transitions one after the other in sequence. |
| Transition | Base class representing an animated transition. |
| Transitioner | Parallel transition with convenience methods for adding new object tweens, helping to incrementally construct a group of transitions. |
| TransitionEvent | Event fired when a Transition starts, steps, ends or is canceled. |
| Tween | Transition that interpolates (in-betweens) properties of a target object over a time interval. |

## Package flare.display

The flare.display package provides classes for display objects that automatically redraw themselves as needed.

|  |  |
| --- | --- |
| **Class** | **Description** |
| DirtySprite | A Sprite that redraws itself as needed when marked as "dirty". |
| LineSprite | A Sprite representing a line. |
| RectSprite | A Sprite representing a rectangle shape. |
| TextSprite | A Sprite representing a text label. |

## Package flare.query

The flare.query package provides classes for creating and processing SQL-style queries over ActionScript objects.

|  |  |
| --- | --- |
| **Class** | **Description** |
| AggregateExpression | Base class representing an aggregate query operator. |
| And | Expression operator that computes the logical "and" of sub-expression clauses. |
| Arithmetic | Expression operator for arithmetic operations. |
| Average | Aggregate operator for computing the average of a set of values. |
| BinaryExpression | Base class for binary expression operators. |
| Comparison | Expression operator for comparing sub-expression values. |
| CompositeExpression | Base class for expressions with an arbitrary number of sub-expressions. |
| Count | Aggregate operator for counting the number of items in a set of values. |
| DateUtil | Utility class providing functions for manipulating Date objects. |
| Distinct | Aggregate (group-by) operator for counting the number of distinct values in a set of values. |
| Expression | Base class for query expression operators. |
| ExpressionIterator | The ExpressionIterator simplifies the process of traversing an expression tree. |
| Fn | Expression operator that performs function invocation. |
| If | Expression operator for an if statement that performs conditional execution. |
| IsA | Expression operator that type checks a sub-expression. |
| Literal | Expression operator for a literal value. |
| Match | Expression operator for text matching operations. |
| Maximum | Aggregate operator for computing the maximum of a set of values. |
| Minimum | Aggregate operator for computing the minimum of a set of values. |
| Not | Expression operator that returns the logical "not" of a sub-expression. |
| Or | Expression operator that computes the logical "or" of sub-expression clauses. |
| Query | Performs query processing over a collection of ActionScript objects. |
| Range | Expression operator that tests if a value is within a given range. |
| StringUtil | Utility class providing functions for manipulating String objects. |
| Sum | Aggregate operator for computing the sum of a set of values. |
| Variable | Expression operator that retrieves a value from an object property. |
| Variance | Aggregate operator for computing variance or standard deviation. |
| Xor | Expression operator that computes the exclusive-or ("xor") of sub-expression clauses. |

## Package flare.scale

The flare.vis.scale package provides classes for data scales such as linear, logarithmic, and ordinal scales.

|  |  |
| --- | --- |
| **Class** | **Description** |
| LinearScale | Scale that spaces values linearly along the scale range. |
| LogScale | Scale that performs a log transformation of the data. |
| OrdinalScale | Scale for ordered sequential data. |
| QuantileScale | Scale that organizes data into discrete bins by quantiles. |
| QuantitativeScale | Base class for representing quantitative numerical data scales. |
| RootScale | Scale that performs a root transformation of the data. |
| Scale | Base class for all data scale types. |
| ScaleType | Constants defining known scale types, such as linear, log, and date/time scales. |
| TimeScale | Scale for timelines represented using Date values. |

## Package flare.util

The flare.util package provides classes for manipulating common data types and accessing object properties in a generalized fashion.

|  |  |
| --- | --- |
| **Class** | **Description** |
| Arrays | Utility methods for working with arrays. |
| Colors | Utility methods for working with colors. |
| Dates | Utility methods for working with Date instances. |
| Displays | Utility methods for working with display objects. |
| Filter | Utility methods for creating filter functions. |
| Geometry | Utility methods for computational geometry. |
| Maths | Utility methods for mathematics not found in the Math class. |
| Orientation | Constants defining layout orientations. |
| Property | Utility class for accessing arbitrary property chains, allowing nested property expressions (e.g., x.a.b.c or x.a[1]). |
| Shapes | Utility class defining shape types and shape drawing routines. |
| Sort | Utility class for sorting and creating sorting functions. |
| Stats | Utility class for computing statistics for a collection of values. |
| Strings | Utility methods for working with String instances. |

## Package flare.util.palette

The flare.util.palette package provides classes for color, shape, and size palettes for visual encoding.

|  |  |
| --- | --- |
| **Class** | **Description** |
| ColorPalette | Palette for color values, including utility methods for generating both categorical and ordinal color palettes. |
| Palette | Base class for palettes, such as color and size palettes, that map from interpolated scale values into visual properties |
| ShapePalette | Palette for shape values that maps integer indices to shape drawing functions. |
| SizePalette | Palette for size values represeneted as scale factors. |

## Package flare.vis

The flare.vis package provides classes for creating interactive data visualizations.

|  |  |
| --- | --- |
| **Class** | **Description** |
| Visualization | The Visualization class represents an interactive data visualization. |

## Package flare.vis.controls

The flare.vis.controls package provides classes for interacting with data visualizations.

|  |  |
| --- | --- |
| **Class** | **Description** |
| AnchorControl | Interactive control for updating a layout's anchor point in response to mouse movement. |
| ClickControl | Interactive control for responding to mouse clicks events. |
| Control | Base class for interactive controls. |
| ControlList | A ControlList maintains a sequential chain of controls for interacting with a visualization. |
| DragControl | Interactive control for dragging items. |
| ExpandControl | Interactive control for expaning and collapsing graph or tree nodes by clicking them. |
| HoverControl | Interactive control for responding to mouse hover events. |
| PanZoomControl | Interactive control for panning and zooming a "camera". |
| SelectionControl | Interactive control for selecting a group of objects by "rubber-banding" them with a rectangular section region. |
| TooltipControl | Interactive control for displaying a tooltip in response to mouse hovers exceeding a minimum time interval. |

## Package flare.vis.legend

The flare.vis.legend package provides classes for presenting legends for data visualizations.

|  |  |
| --- | --- |
| **Class** | **Description** |
| Legend | A legend describing the visual encoding of a data property. |
| LegendItem | An item in a discrete legend consisting of a label and an icon indicating color, shape, and/or size. |
| LegendRange | A range in a continuous legend, consisting of a continuous visual scale and value labels. |

## Package flare.vis.operator.label

The flare.vis.operator.label package provides classes for creating and positioning labels.

|  |  |
| --- | --- |
| **Class** | **Description** |
| Labeler | Labeler that adds labels for items in a visualization. |
| RadialLabeler | Labeler that positions labels around a circle in polar coordinates. |
| StackedAreaLabeler | Labeler for a stacked area chart. |

## Package flare.vis.operator.layout

The flare.vis.operator.layout package provides classes for calculating the spatial position of visualized data.

|  |  |
| --- | --- |
| **Class** | **Description** |
| AxisLayout | Layout that places items along the X and Y axes according to data properties. |
| BundledEdgeRouter | Layout that routes edges in a graph so that they form groups, reducing clutter. |
| CircleLayout | Layout that places items in a circular layout. |
| CirclePackingLayout | Layout that places nodes as circles compacted into a larger circle. |
| DendrogramLayout | Layout that places items in a dendrogram for displaying the results of hierarchical clustering. |
| ForceDirectedLayout | Layout that places nodes based on a physics simulation of interacting forces. |
| IcicleTreeLayout | Layout that places nodes in an icicle layout, distributing nodes evenly within the display bounds. |
| IndentedTreeLayout | Layout that places tree nodes in an indented outline layout. |
| Layout | Base class for all operators that perform spatial layout. |
| NodeLinkTreeLayout | Layout that places nodes using a tidy layout of a node-link tree diagram. |
| PieLayout | Layout that places wedges for pie and donut charts. |
| RadialTreeLayout | Layout that places tree nodes in a radial layout, laying out depths of a tree along circles of increasing radius. |
| RandomLayout | Layout that places nodes randomly within the layout bounds. |
| StackedAreaLayout | Layout that consecutively places items on top of each other. |
| TreeMapLayout | Layout that places node in a TreeMap layout that optimizes for low aspect ratios of visualized tree nodes. |