



JViews Diagrammer glossary

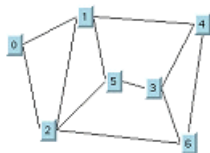
Table of contents

JViews Diagrammer Glossary.....5

JViews Diagrammer Glossary

2-connected graph

(Graph Layout) Also known as a biconnected graph: A *connected graph* that will remain a connected graph if any one node is removed. (In other words, a graph in which at least two distinct *paths* exist between every two nodes.)



about-to-change event

When a property value is about to be notified to the model, it is advisable to notify interested listeners as well so that they have an opportunity to constrain or even to veto the proposed new property value.

acyclic graph

(Graph Layout) A graph that has no *cycles*.

adjustment session

A sequence of changes defined between `setXYZAdjusting(true)` and `setXYZAdjusting(false)` statements. Adjustment sessions are used to indicate to listeners of the changes that the changes are part of a sequence. This allows to optimize listeners to validate the changes only at the end of the adjustment session, when the `setAdjusting(false)` statement is issued. The performance of an application can improve dramatically if adjustment sessions are used in the proper way. Examples of changes that can be included in an adjustment session are: changes to a style sheet, to the contents of a manager, to the SDM model, and many more.

annealing (simulated)

(Graph Layout) A mathematical technique used to place links in such a way that they do not overlap with one another nor with the nodes that they interconnect.

applet

A program designed to be executed from within an application and not directly from the operating system. An applet is usually a simple, single-function program that ships with a larger product.

automatic layout

(Graph Layout) A layout process where the *layout algorithm* does everything without any user intervention.

biconnected graph

See *2-connected graph*.

bundle

See *link bundle*.

bus topology

(Graph Layout) A type of network topology where a set of nodes is connected to a bus object.

colinear

(Hypergraph) Segments in a *hyperedge* that have the same angle. Colinear segments should not be connected.

combinator

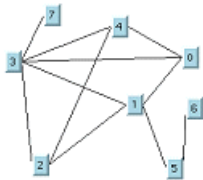
(CSS) A single character that may separate two selectors.

connected component

(Graph Layout) A connected graph or subgraph. A connected component of a flat graph G is a maximal connected subgraph of G .

connected graph

(Graph Layout) A graph where there is a path connecting each pair of nodes. See also *disconnected graph*.



connection

(Graph Layout) Another name for an edge of a graph. See also *edge*.

CSS

(CSS) Cascading Style Sheets. A mechanism for adding style, such as fonts, colors, spacing, to Web documents. The CSS language is specified by World Wide Web Consortium (W3C) Recommendations. Full Recommendations exist for CSS level 1 and level 2.

CSS class

(CSS) An optional part of a *selector*. A CSS class is a word preceded by a dot, which represents a *user-defined type* to be used in *pattern-matching*.

CSS for Java

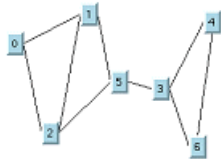
(CSS) A proprietary translation of CSS2 by IBM® ILOG® to make it available for styling Java™ objects.

CSS2

(CSS) Cascading Style Sheets, level 2. A style sheet language that allows authors to attach style to structured documents, such as HTML documents or XML applications. CSS2 separates the presentation style from the content and thus simplifies Web authoring and site maintenance.

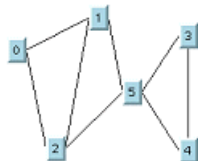
cut-edge

(Graph Layout) An edge is a cut-edge if the graph is no longer connected when the edge is removed from the graph. In the following illustration, the link between nodes 3 and 5 is a cut-edge. Note that its end points are cut-nodes.



cut-node

(Graph Layout) A node is a cut-node if the graph is no longer connected when the node is removed from the graph. In the following illustration, node 5 is a cut-node.



cycle

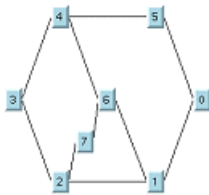
(Graph Layout) A path of a graph that begins and ends on the same node. Also called a loop.

cyclic graph

(Graph Layout) A graph that contains cycles.

cyclomatic number

(Graph Layout) A number equal to $m - n + 1$, where m = number of edges and n = number of nodes. In the following illustration, the number of edges = 10, the number of nodes = 8. Therefore, the cyclomatic number is $10 - 8 + 1 = 3$.



dashboard

In IBM® ILOG® JViews Diagrammer, an instance of the class `IlvDashboardDiagram`. It is an extension of the *diagram component* that manages a collection of *symbols* and *links*. Use the *Dashboard Editor* to develop dashboards and the *Dashboard API* to display dashboards in Swing applications.

Dashboard API

In IBM® ILOG® JViews Diagrammer, the `ilog.views.dashboard` package. This collection of classes is used to develop *Dashboard Editor* applications and add *dashboards* to Swing applications.

Dashboard Editor

The GUI supplied for developing a *dashboard*.

data model

The interface with which business objects must comply to be represented in a diagram. The data model may make use of your business objects directly or require a transformation of them. One major feature of IBM® ILOG® JViews Diagrammer is that the data model is separated from the visualization part.

data source

A container of data supported by IBM® ILOG® JViews Diagrammer, such that all or part of the data can be loaded into the Designer and used in a diagram. Examples of data sources are a Microsoft® Access® database, a CSV (comma-separated values) file, an XML file in diagram format.

declaration

(CSS) The elements of the right side of a *style rule*. The declaration set is enclosed within curly brackets. Each declaration is a property-value pair. Each property-value pair ends with a semi-colon. A property-value pair sets a rendering *property* on a graphic object that represents a model object.

degree

The degree of a node v of a graph G is the number of graph edges which touch v .

Designer

The GUI supplied for developing a *diagram component*.

diagram component

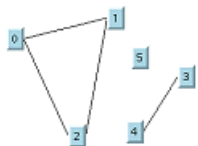
In IBM® ILOG® JViews Diagrammer, an instance of the class `IlvDiagrammer`. This is a Swing component used to manage a diagram. It contains a grapher that manages a collection of nodes and links, and a view to display the diagram. The nodes in a diagram are represented by *symbols*.

directed graph

(Graph Layout) A graph in which every edge is associated with an ordered pair of nodes. Also called an oriented graph.

disconnected graph

(Graph Layout) A graph containing at least two nodes that are not linked by a path. In the following illustration, node 5 is not linked by a path to any other node. Nodes 3 and 4 are not linked by a path to any nodes other than nodes 3 and 4.



double buffering

The rendering technique of first drawing to an offscreen image (back buffer), and when the image is complete, copying the offscreen image to the screen (front buffer). In applets, drawing is almost always done with double buffering to reduce flickering. See also *triple buffering*.

edge

(Graph Layout) A line connecting two vertices in a graph. An edge is also called a link or a connection. In this documentation, the term link is primarily used for the term edge.

edge crossing

See *link crossing*.

element

(CSS) The primary syntactic construct of a document language. Most CSS style sheet rules use the name of an element (such as <P>, <TABLE>, for HTML) to specify rendering information. In CSS for Java™ objects, the role of element is attributed to the term *model object*."

end segment

(Hypergraph) A segment in a segmented *hyperedge* that is *incident* to a node.

fixed link

(Graph Layout) A link which remains in the same position during graph layout or link layout. If the user specifies a link as fixed, layout algorithms are not allowed to reshape this link.

fixed node

(Graph Layout) A node which remains in the same position during graph layout or link layout. If the user specifies a node as fixed, layout algorithms are not allowed to move this node.

flat graph

(Graph Layout) The converse of a nested graph: a graph of which no node is itself a graph. *See nested graph.*

force-directed layout

(Graph Layout) A class of layout algorithms for creating straight-line drawings of *undirected graphs* based on an iterative computation of the position of the nodes according to a set of attractive and repulsive “forces.” These forces are computed in such a way that they tend to produce a layout with only a few link crossings or no link crossings. The Uniform Length Edges Layout of JViews Diagrammer is an example.

gesture

A series of one or more events triggered by the end user.

graph

(Graph Layout) A finite set of nodes (also called vertices) connected by a finite set of links (also called edges or connections).

graph layout

(Graph Layout) The process that applies a *layout algorithm* to a graph. Also the graph drawing that results from the layout process.

graph model

(Graph Layout) In the context of graph layout, a Java™ class that defines a suitable, generic API for graphs or an instance of the class `IlvGraphModel`.

grapher

In IBM® ILOG® JViews Framework, an instance of the class `IlvGrapher`. More generally, a Java™ object that is used to manage a collection of nodes and links.

grid drawing

(Graph Layout) A drawing where nodes and link bends have discrete (integer) coordinates.

hyperedge

(Hypergraph) Link in a *hypergraph* that has multiple sources and multiple sinks. A hyperedge connects multiple nodes. Hyperedges are used in applications such as electrical signal diagrams, multiframe visualization, network management, and UML diagrams.

hypergraph

(Hypergraph) A graph that contains *hyperedges*. More specifically, it contains a finite set of nodes (also called vertices) connected by a finite set of hyperedges.

hypergrapher

(Hypergraph) An instance of the class `IlvHyperGrapher`. It is a grapher that can contain three types of object: *node*, *link*, and *hyperedge*.

image map

Images on the client side with an attached map that points out certain hot spots or clickable areas. They are typically used for displaying tooltips.

incident

(Graph Layout) A link-node connection. A link is incident to a node if the node is at one end of the link. A node is incident to a link if that link is incident to the node.

incremental layout

(Graph Layout) A layout process where the result of a previous layout is used as the starting point for applying the layout algorithm a second time to a modified graph in order to minimize the changes.

inner segments

(Hypergraph) The segments in a segmented *hyperedge* that are not *incident* to a node.

interactor

An agent for a graphic object that allows the user to obtain specific behavior from the object concerned.

intergraph hyperedge

(Hypergraph) In a nested *hypergraph*, a *hyperedge* with end nodes in different hypergraphs.

intergraph link

(Graph Layout) In a nested graph, a link whose end nodes are contained in different subgraphs.

JSF

JavaServer™ Faces.

JSP

JavaServer™ Pages.

layer

In IBM® ILOG® JViews Framework, the storage area of a *manager* in which graphic objects are placed. A manager uses multiple layers, referenced by index numbers. Objects in a higher-numbered layer are displayed in front of objects in a lower-numbered layer.

layout algorithm

(Graph Layout) The process that computes new coordinates for nodes and/or new shapes for links in order to obtain a suitable representation of a graph.

layout region

(Graph Layout) In the context of graph layout, the rectangle where the graph drawing will be placed when the graph is laid out.

link

(Graph Layout) Another name for an edge in a graph. In this documentation, the term link is primarily used for edge. See also *edge*.

link bundle

(Graph Layout) A set of links (edges) connecting a given pair of nodes in a graph, drawn as a set of parallel lines. See *multiple links*.

link connector

(Graph Layout) In IBM® ILOG® JViews Diagrammer, the system that allows you to control the connections of links to nodes in a grapher.

link crossing

(Graph Layout) Link crossings occur when links intersect at places other than an incident node. Also called edge crossings. Often, layout algorithms are used to minimize the number of link crossings.

listener

In IBM® ILOG® JViews Diagrammer, an interface allowing you to connect events that occur in the manager view with actions to be performed.

loop

See *cycle*.

manager

In IBM® ILOG® JViews Framework, a holding place for grouping graphic objects and for coordinating their behavior and display in multiple layers and multiple views.

manager view

See *view*.

model objects

The application objects in an SDM *data model*.

multiple links

(Graph Layout) More than one link between the same origin and destination nodes.

neighbor nodes

(Graph Layout) The nodes that are connected to a given node by a link.

nested graph

(Graph Layout) A graph that contains nodes that are graphs, that is, a graph that contains nested subgraphs.

node

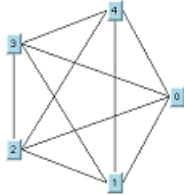
(Graph Layout) Another name for a vertex of a graph. In this documentation, the term node is primarily used for vertex.

nonplanar graph

(Graph Layout) A graph that cannot be drawn without any links crossing other links.

nonzoomable

See *zoomable*.



NP-complete

(Graph Layout) A class of computational problems for which no efficient solution algorithm has been found. Many significant computer science problems, including many graph layout problems, belong to this class.

orthogonal drawing

(Graph Layout) A drawing where each link is drawn as a polygonal chain of alternating horizontal and vertical segments.

palette

A set of items from which to select, typically used in a graphical application. Examples are a color palette for assigning a color to a graphic object, an object palette for building a composite graphic object.

pan

To pan is to move a display such as a diagram or map within a GUI window or pane, in this case, the Diagram pane. The purpose of panning is to control which portion of the diagram or map is visible.

path

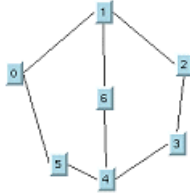
(Graph Layout) A sequence of consecutive nodes leading from one node to another using the links of the graph. It is the route along the links through the nodes in a graph. The length of a path is the number of links traversed.

pattern-matching

(CSS) The process of analyzing a *selector* to find the model objects to which the *style rule* applies.

planar graph

(Graph Layout) A graph that can be drawn with no links crossing other links.



polyline drawing

(Graph Layout) A drawing where each link is drawn as a polygonal chain.

preferred layout

(Graph Layout) In the context of the Recursive Layout, the layout instance set as the one to be used for each graph. It is stored in a property of the graph.

project

The combination of a *style sheet* and a *data source*, identified in a project file (.idpr file) in XML format.

property

(CSS) A named characteristic of a graphic object to which you can assign values.

pseudo-class

(CSS) A CSS construct in a *selector*. A pseudo-class is a word preceded by a colon (:). It represents a state given by the execution context and not by the underlying data model nor by the structure of the document.

pseudo-element

(CSS) A CSS construct in a *selector*. A pseudo-element is a word preceded by a colon (:). It represents a state given by the structure of the document and not by the underlying data model nor by the execution context.

radial drawing

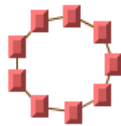
(Graph Layout) A layout style where the nodes are placed radially around a root node.

rich client

(or fat client or thick client) A way of deploying a client/server application such that the client performs most of the processing itself rather than depending on the server facilities. All users have to install and maintain their own copy of the program but may be able to do useful work when disconnected from the server. User response can be quicker and more intelligent due to application state existing locally ("model" as well as "view"). See also *thin client*.

ring

(Graph Layout) A type of network topology where the nodes are arranged in a circular configuration.



SDM model

See *data model*.

seed value

The value that is used for the initialization of a random number generator. Some layout algorithms use random numbers during the layout computation.

selection box

A dashed box that appears around nodes or links when they are selected in the *Designer*.

selector

(CSS) The left side of a *style rule*. A selector consists of a mandatory object type, which is a class in a data model, and optional tags, pseudo-classes, pseudo-elements, and

attribute conditions. A selector undergoes *pattern-matching* to determine the scope of the *style rule*.

self-link

(Graph Layout) A link whose origin and destination nodes are the same node.

semiautomatic layout

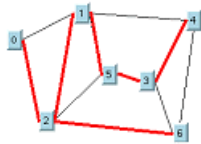
(Graph Layout) A layout process where the user makes manual improvements to the result of the automatic layout process.

separable model architecture

A Swing variant of the traditional MVC design of user-interface objects, where the view and controller parts are bundled together.

spanning tree

(Graph Layout) A minimal subgraph, defined as follows: A spanning tree *S* of a flat graph *G* is a subgraph of *G* containing all the nodes of the graph and whose links are a subset of the links of the graph. The number of links of *G* that are not present in *S* must be the minimum number for which there are no cycles in *S*. The spanning tree is shown by the red links in the following illustration.

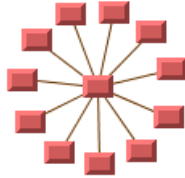


spline

(Graph Layout) One possible appearance of a link connecting nodes: it is a smooth curve governed by control points.

star

(Graph Layout) A type of network topology where the nodes are arranged on a circle with each node being connected to a center node.



static layout

(Graph Layout) A layout process where the layout algorithm is completely redone each time the graph is changed.

straight-line

(Graph Layout) Denotes a drawing where each link is drawn as a straight line segment.

style file

(CSS) A file containing CSS statements.

style rule

(CSS) A formal statement which conforms to the CSS syntax and is used to attach styles to model objects. A style rule contains a *selector* and a *declaration*.

style sheet

(CSS) A list of style rules for customizing the appearance of a diagram in a *diagram component*. A style sheet is saved in a file with extension `.css`.

subgraph

(Graph Layout) A graph that is contained in another graph. In flat graphs, G' is a subgraph of G if its node and link sets are included in the node and link sets of G .

In nested graphs, a node that is a graph is called a subgraph of the nested graph.

subobject

(Styling) An object that is created during the application of a declaration in a CSS style sheet, by virtue of a declaration value of the form `@#identifier`, `@=identifier`, or `@+identifier`. It is styled through the same CSS style sheet, although it is not contained in the original CSS model.

swim lane

Swim lanes (also known as the Rummler-Brache approach) provide a way of imposing organizational structure on processes. The lanes, which are organizational bands representing different participants or departments in an organization, are placed around subsets of the process steps. The swim lane enclosing a set of steps delineates the ownership of those steps for process management purposes.

SWT

The Standard Widget Toolkit (*SWT*) is the window toolkit of the Eclipse™ development environment and the Eclipse Rich Client Platform (RCP).

symbol

A symbol is a collection of graphic objects, parameters and conditions used to give a dynamic representation of changing data. It is made up of objects and other symbols. Symbols are created using the *Symbol Editor*. A finished symbol is manipulated by being imported into other IBM® ILOG® JViews applications such as the *Designer*, the *Dashboard Editor*, or directly using the IBM® ILOG® JViews Java™ API.

Symbol Editor

The GUI supplied for developing *symbols*.

terminated by a segment

(Hypergraph) A segment of a *hyperedge* where another segment with variable angle starts or ends. The segment terminates the segment with variable angle.

theme

A *style sheet* that gives a particular look-and-feel. It is available as a starter style sheet.

thin client

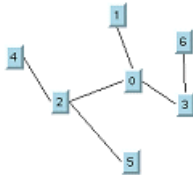
(or light client or clientless) A thin client is a way of deploying a client/server application such that the client side is small, with no Java™ running on the client. Users require only a Web browser and an Internet connection. The user interface is executed within the browser with no local system administration and the business logic is executed on a remote server. See also *rich client*.

topology

(Graph Layout) The structure of a graph. Two drawn graphs have the same topology if you can obtain one drawing from the other by moving the nodes and reshaping the links.

tree

(Graph Layout) An undirected tree is a connected undirected *acyclic graph* (that is, a graph that does not contain any cycles). A directed tree is a connected directed graph where each node has exactly one incoming link except the root node, which has no incoming links.



triple buffering

A rendering technique that uses three frame buffers instead of two as in double buffering. With two buffers, front and back, the front buffer is being displayed while the back buffer is being drawn. When the back buffer is ready, it has to wait until the current scan of the screen is finished before the buffers can be swapped. With three buffers, there is no waiting time. Triple buffering results in smoother animation, especially if the screen has a slow refresh rate. See also *double buffering*.

undirected graph

(Graph Layout) A graph in which every link is associated with an unordered pair of nodes.

user-defined type

A type that is not necessarily defined in a class of the data model but is recognized implicitly from the data in an XML data file or JDBC database that has been read into JViews Diagrammer. In CSS syntax, the user-defined type appears as a *CSS class*.

vertex

(Graph Layout) A “dot” in a graph. A graph consists of a finite set of vertices connected by a finite set of edges (also called links or connections). A vertex is also called a node. In this documentation, the term node is primarily used for the term vertex.

view

In IBM® ILOG® JViews, the AWT or Swing component where graphic objects are displayed. To display graphic objects contained in the different *layers* of a *grapher*, you create at least one view, and often multiple views. The grapher lets you connect as many views as you require to display graphic objects.

zoomable

A graphic object is said to be *zoomable* if its bounding box follows the zoom level. Otherwise, the object is *nonzoomable*.