

## ***ILOG JViews TGO Glossary***

<b>adapter</b>	An adapter converts business objects into representation objects and pushes these objects into the corresponding representation model. There is a specific adapter for each representation model (that is, table, tree, equipment, and network). An adapter is generally connected to a data source.
<b>Bellcore state dictionary</b>	A dictionary of telecommunication object states based on the Bellcore (Bell Communications Research) standard specified in <i>GR-1093 — Generic State Requirements for Network Elements</i> .
<b>blinking manager</b>	A context service that manages the blinking of colors.
<b>BTS</b>	A predefined business object. A BTS (Base Transceiver Station) is a base station composed of antennas that relay (receive and transmit) radio messages within cells of a cellular phone system.
<b>business model</b>	The business model describes the attributes and inheritance of business object classes. It serves as a common vocabulary between ILOG JTGO and the back-end application. It does not address relationships between these classes.
<b>business object</b>	An instance of a business class. A business object typically corresponds to a “real” business object somewhere in the back-end application. Business objects represent a subset of the back-end data manipulated by ILOG JTGO. They can be shared across multiple graphic components. ILOG JTGO furnishes a number of predefined business objects.  See business model. See also the predefined business objects: BTS, card, group, link, network element, off-page connector (OPC).
<b>card</b>	A predefined business object. An item of equipment that is inserted in a shelf.

<b>card carrier</b>	A predefined business object. A card carrier is a container for shelf item objects. Card carriers allow you to associate more than one card with a single slot.
<b>class manager</b>	A context service that handles a business model. The class manager makes business object classes available to the whole application and allows them to be used by all of the components.
<b>computed attribute</b>	A computed attribute is an attribute of which the value is derived from the value of other attributes. For example, given a start and a finish date, it is possible to calculate a duration.
<b>context</b>	The application context provides access to common contextual data (such as the locale or the current user) and services (such as the URL access service or the class manager).
<b>controller</b>	The controller translates interactions with the view, such as mouse clicks and keystrokes, into updates to be performed on the representation model, instructing the model and/or the view to change as appropriate.
<b>CSS</b>	Cascading Style Sheets. A language that provides rich stylistic control of content to be published on browsers.
<b>CSS2</b>	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.
<b>data source</b>	A data source retrieves business objects from a back-end application and transforms them into <code>ILpObject</code> instances. A data source is back-end specific, which means that there is a data source for each type of back end to be addressed.
<b>data source manager</b>	A context service that provides access to the data sources that it handles.
<b>declaration</b>	(CSS.) A property and a corresponding value.
<b>deployment descriptor</b>	A file that defines the application behavior. It is used to initialize an application context.
<b>dictionary visuals</b>	A set of predefined visuals to graphically represent state and alarm dictionaries based on telecommunication standards. A set of attributes associated with predefined styles and rendering policies.
<b>dynamic attribute model</b>	The dynamic attribute model allows you to define the attributes carried by an object at run time.

See static class.

<b>empty slot</b>	A predefined business object. An empty portion of a shelf that can represent alarms and states graphically.
<b>equipment component</b>	The equipment component is one of the four graphic components provided in ILOG JTGO. It is designed to display and interact over telecommunication equipment such as shelves, cards, ports, links and LEDs. The equipment component encapsulates the MVC (Model View Controller) architecture.
<b>equipment editor</b>	The equipment editor is a graphical user interface which allows you to design and edit equipment elements such as shelves, cards, ports and LEDs.
<b>expansion strategy</b>	A strategy used at the adapter level to identify whether objects should be loaded or not in the representation model. The expansion strategy defines how an object is going to behave when it is expanded. It also indicates whether load on demand is implemented.
<b>graphic component</b>	ILOG JTGO graphic components encapsulate the MVC (Model View Controller) triad that clearly separates data from its graphic representation. ILOG JTGO provides four main components: the table component, the tree component, the network component, and the equipment component.  See also controller, representation model, view.
<b>graphic holder</b>	A graphic holder is responsible for storing the graphic objects created for a given graphic view. In simple cases, where graphic objects can be recreated when needed, it may not be necessary to store them in a graphic holder.
<b>graphic object</b>	Graphic objects are the graphic representation of their associated representation objects in a graphic view. Graphic objects are optional. They are generally used to draw complex composite objects. Graphic objects are specific to a graphic view, but certain graphic object classes can be shared across graphic components.
<b>graphic view</b>	A graphic view delimits a portion of the screen where the representation objects held by the model are drawn.  See graphic object, view.
<b>group</b>	A predefined business object. A group is a container that holds a set of network elements or links. Groups are used to present a set of network resources grouped logically or geographically.
<b>handler</b>	The handler is an optional way of controlling the data source and adapters. When used, the handler is responsible for handling requests coming from its

	associated graphic component that affect back-end data.
	See equipment component, network component.
<b>image repository service</b>	A context service used to load all ILOG JTGO images.
<b>impact alarm</b>	Impact alarms are used in combination with alarm propagation. An impact alarm corresponds to a propagated alarm, reported by a network element, but carried by another element.
<b>interactor</b>	Interactors translate user gestures and keystrokes into actions to be performed, and more specifically into Swing action invocations. ILOG JTGO provides two types of interactor: view interactors that operate on the view as a whole and object interactors that apply to each object contained in the view individually.
<b>JNDI</b>	Java Naming and Directory Interface. JNDI is an API specified in Java that provides naming and directory functionality to applications written in Java. Naming and directory services provide network-wide sharing of a variety of information about users, machines, networks, services, and applications.
<b>LED</b>	A predefined business object. An LED (Light-Emitting Diode) shows the condition of an item of hardware or software by emitting a colored light.
<b>link</b>	A predefined business object. A link represents a physical connection between two network elements. Links are used to display the transmission elements making up the network lines.
<b>link bundle</b>	A predefined business object. The link bundle object is a link container that you can collapse or expand.
<b>link factory</b>	The link factory transforms business objects that are links into representation objects that are instances of <code>IlpNetworkLink</code> .  See node factory.
<b>link set</b>	A predefined business object. A link set is used to represent a collection of links between network resources, laid out in a specific order with a specific distance.
<b>load on demand</b>	A feature supported by the network, the equipment and the tree components. It is implemented according to the expansion strategy. It means that the graphic representation of a business object is only created when the parent object is expanded.
<b>Misc state dictionary</b>	A dictionary of secondary states that can be used to complement OSI,

Bellcore or SNMP standards.

See Bellcore state dictionary, OSI state dictionary SNMP state dictionary, SONET state dictionary.

- model** Stores the application objects to be represented in the view. In ILOG JTGO, this role is carried out by the representation model.
- See MVC (Model View Controller).
- MVC (Model View Controller)** The Model View Controller architecture is an object-oriented pattern used in user interface design to separate the application object (model) from the way it is represented to the user (view) and from the way in which the user controls it (controller).
- See controller, representation model, view.
- network component** The network component is one of the four graphic components supplied with ILOG JTGO. It can display all kinds of ILOG JTGO objects: network elements, links, groups, polylines, off-page connectors, cards, card carriers, shelves, ports, LEDs. The network component encapsulates the MVC (Model View Controller) architecture.
- network element** A predefined business object. A network element is any kind of shelf-based telecom or data-communication equipment (such as a switch, a multiplexer, a cross connect system), outside plant equipment (such as a coax nodes), or peripheral equipment (terminal or printer).
- node factory** The node factory transforms business objects other than links into representation objects that are instances of `IlpNetworkNode`.
- See link factory.
- object interactor** An object interactor listens to all user events occurring on a graphic object and sets off the corresponding Swing actions.
- See interactor.
- off-page connector (OPC)** A predefined business object. An off-page connector can be used in place of a network node to show that the incoming link extends outside the displayed view.
- OSI state dictionary** A dictionary of telecommunication object states based on the OSI (Open System Interconnection) standard specified in *ISO/IEC 10164-2, ITU-T X.731 — State Management Function*.

<b>outstanding alarms</b>	Outstanding alarms on a managed object include the new and the acknowledged alarms but not the cleared alarms.
<b>partial network element</b>	A partial network element is an abstraction which denotes a network element that is only part of the real-world network element.
<b>passive device</b>	A predefined business object. A passive device does not have a specific object state and does not report information about its current states and alarms.
<b>Performance state dictionary</b>	<p>A dictionary of secondary state values that can be used to complement any other standard state system such as OSI, Bellcore, SNMP, or SONET.</p> <p>See Bellcore state dictionary, OSI state dictionary, SNMP state dictionary, SONET state dictionary.</p>
<b>plinth</b>	A relief rectangle added to a group or link to show the alarm count and possibly additional information.
<b>port</b>	A predefined business object. A port connects a card to an external device.
<b>predefined business object</b>	A business object provided by ILOG JTGO and specifically designed to help you build high-quality, user-friendly user interfaces in the domain of telecom network management.
<b>property</b>	(CSS.) A stylistic parameter that can be implemented through CSS.
<b>pseudoclass</b>	Used in CSS selectors to allow information external to the source, such as whether an anchor has been visited or not, to classify elements.
<b>pseudoelement</b>	Used in CSS selectors to handle typographical items rather than structural items.
<b>raw alarm</b>	An alarm reported by a network element and carried by this element.
<b>renderer</b>	A renderer draws a graphic object based on the information retrieved from its associated representation object and business object, if any, using graphic specifications from styles and visual dictionaries. There are two types of renderer: those that implement the default Swing renderer interfaces (only for table and tree) and those that build a graphic object.
<b>representation model</b>	A representation model is a container for representation objects. It is used to produce the graphic objects in the view. The representation model manages the behavior and data of the application domain, responds to requests for information about its state (coming usually from the view), and responds to instructions to change its state (usually issued by the controller).

<b>representation object</b>	<p>A representation object provides the minimal data and structure information required for generating graphic objects, but is totally independent of a particular graphic rendering. More specifically, it does not contain graphic-related information. A representation object is specifically designed to work with a particular graphic component (table, tree, network, and so on) for which it provides a default implementation. A representation object can be linked to a business object, if any, but not necessarily.</p> <p>See representation model.</p>
<b>rule</b>	(CSS.) A selector and its declarations.
<b>SAN state dictionary</b>	<p>A dictionary of secondary state values that can be used to complement any other standard state system such as OSI, Bellcore, SNMP, or SONET.</p> <p>See Bellcore state dictionary, OSI state dictionary, SNMP state dictionary, SONET state dictionary.</p>
<b>selector</b>	A condition that is matched against objects from the business model. Selectors allow you to customize objects according to their business class or attributes.
<b>shelf</b>	A predefined business object. A part of a shelf-based telecommunication item of equipment used to hold cards.
<b>shortcut network element</b>	A shortcut network element is an abstraction denoting an object that is only a reference to an existing network element.
<b>SNMP state dictionary</b>	A dictionary of telecommunication object states based on the SNMP (Simple Network Management Protocol) standard specified in <i>RFC 1213 — Management Information Base for Network Management of TCP/IP based internets: MIB-II</i> .
<b>SONET state dictionary</b>	A dictionary of transport link states based on the SONET (Synchronous Optical Network) standard.
<b>specificity</b>	The weighting of the priority of a rule in CSS in accordance with the number of components in its selector.
<b>state dictionaries</b>	To describe the state and alarm conditions of a telecom object, ILOG JTGO provides several state dictionaries based on worldwide standards. The development of these dictionaries is based on current telecom standards combined with hands-on experience.
<b>static class</b>	<p>A Java class defined at compilation time.</p> <p>See dynamic attribute model.</p>

<b>style sheet</b>	<p>A collection of rules for controlling the presentation of Web pages in a precise manner.</p> <p>See CSS2.</p>
<b>subnetwork</b>	<p>A subnetwork allows you to create applications that display a network inside another network. It is created automatically when you define a containment relationship between objects in the data source. A subnetwork can be displayed collapsed or expanded in the network component.</p>
<b>synchronization strategy</b>	<p>A context service that makes it possible to access shared resources simultaneously with a choice of synchronization schemes.</p>
<b>table component</b>	<p>The table component is one of the four graphic components provided in ILOG JTGO. It displays objects in rows and their attributes in columns. The table component encapsulates the MVC (Model View Controller) architecture.</p>
<b>thin client</b>	<p>A very light client application that can be run inside a web browser.</p>
<b>trap state</b>	<p>A trap represents something unusual that occurs in an object.</p>
<b>tree component</b>	<p>The tree component is one of the four graphic components provided in ILOG JTGO. It displays a hierarchical view of the objects. The tree component encapsulates the MVC (Model View Controller) architecture.</p>
<b>type converter</b>	<p>A context service that is mainly used to read and write XML files that contain data source information. All the attributes of business objects are converted using this service.</p>
<b>URL access service</b>	<p>A context service that turns relative URLs into absolute URLs or pathnames. It can be used to deploy applications and applets without hardcoding pathnames or URLs. It can also be used to access configuration files, data source files and resource files.</p>
<b>view</b>	<p>The view manages the visual display of the data represented by the model.</p> <p>See graphic view, MVC (Model View Controller).</p>
<b>view interactor</b>	<p>A view interactor listens to all user events occurring on a view and dispatches intercepted events to object interactors.</p>