Contents

Chapter 1

Package com.imagos.javaglade.vo

1.1 Classes

1.1.1 Class **GladeAccelerator**

ValueObject with the data of a GladeAccelerator item. <br/>Responsibilities:

- Store the gladeAccelerator data.</li>

<br/>

Declaration

public class GladeAccelerator
extends java.lang.Object

Constructors

- *GladeAccelerator*
  `public` **GladeAccelerator**`(` `int` **key**`,` `java.lang.String` **signal**`,` `java.lang.String` **modifiers** `)`
  - **Usage**
    - Default GladeAccelerator constructor.

- *GladeAccelerator*
  `public` **GladeAccelerator**`(` `int` **key**`,` `java.lang.String` **signal**`,` `java.lang.String` **modifiers** `)`
  - **Usage**
    - GladeAccelerator constructor that initializes the Glade accelerator values.
  - **Parameters**
    - `key` - int, accelerator input key ASCII code.
    - `signal` - String, signal name of the signal linked to the accelerator.
    - `modifiers` - String, accelerator modifiers (for combinations of keys).

- *GladeAccelerator*
  `public` **GladeAccelerator**`(` `java.lang.String` **key**`,` `java.lang.String` **signal**`,` `java.lang.String` **modifiers** `)`
  - **Usage**
    - GladeAccelerator constructor that initializes the Glade accelerator values.
  - **Parameters**
    - `key` - String, accelerator input key.
    - `signal` - String, signal name of the signal linked to the accelerator.
    - `modifiers` - String, accelerator modifiers (for combinations of keys).

METHODS

- *getKey*
  `public int` **getKey**`( )`
  - **Usage**
    - Gets the accelerator input key.
  - **Returns** - int, accelerator input key.

- *getModifiers*
  `public String` **getModifiers**`( )`
  - **Usage**
    - Gets the accelerator modifiers.
  - **Returns** - String, accelerator modifiers.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getSignal*
  `public String` **getSignal**`( )`
  - **Usage**
    - Gets the accelerator signal.
  - **Returns** - String, accelerator signal.

- *setKey*
  `public void` **setKey**`( int` **key** `)`
  - **Usage**
    - Sets up the accelerator input key.
  - **Parameters**
    - `key` - int, accelerator input key.

- *setModifiers*
  `public void` **setModifiers**`( java.lang.String` **modifiers** `)`
  - **Usage**
    - Sets up the accelerator modifiers.
  - **Parameters**
    - `modifiers` - String, accelerator modifiers.

- *setSignal*
  `public void` **setSignal**`( java.lang.String` **signal** `)`
  - **Usage**
    - Sets up the accelerator signal.

    - `signal` - String, accelerator signal.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the accelerator info to a formatted String.
  - **Returns** - String, formatted String with the accelerator's data.

1.1.2 CLASS **GladeAccessibility**

ValueObject with the data of a GladeAccessibility item. <br/>Responsibilities:

- Store the gladeAccessibility data.</li>

<br/>

DECLARATION

public class GladeAccessibility
extends java.lang.Object

CONSTRUCTORS

- *GladeAccessibility*
  `public` **GladeAccessibility**`( )`

METHODS

- *addAtkProperty*
  `public void` **addAtkProperty**`( com.imagos.javaglade.vo.GladeProperty` **prop** `)`
  - **Usage**
    - Adds a new atk property to the accessibility atk properties' list.
  - **Parameters**
    - `prop` - GladeProperty, atk property's data to insert.

- *getLAtkActions*
  `public ArrayList` **getLAtkActions**`( )`
  - **Usage**
    - Gets the atk actions' list.
  - **Returns** - ArrayList<GladeAtkAction>, list of atk actions.

- *getLAtkProperties*
  `public ArrayList` **getLAtkProperties**`( )`

    - Gets the atk properties' list.
  - **Returns** - ArrayList<GladeAtkProperty>, list of atk properties.

- *getLAtkRelations*
  `public ArrayList` **getLAtkRelations**`( )`
  - **Usage**
    - Gets the atk relations' list.
  - **Returns** - ArrayList<GladeAtkRelation>, list of atk relations.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *setLAtkActions*
  `public void` **setLAtkActions**`( java.util.ArrayList` **atkActions** `)`
  - **Usage**
    - Sets up the atk actions' list.
  - **Parameters**
    - `atkActions` - ArrayList<GladeAction>, list of atk actions.

- *setLAtkProperties*
  `public void` **setLAtkProperties**`( java.util.ArrayList` **atkProperties** `)`
  - **Usage**
    - Sets up the atk properties' list.
  - **Parameters**
    - `atkProperties` - ArrayList<GladeAtkProperty>, list of atk properties.

- *setLAtkRelations*
  `public void` **setLAtkRelations**`( java.util.ArrayList` **atkRelations** `)`
  - **Usage**
    - Gets the atk relations' list.
  - **Parameters**
    - `atkRelations` - ArrayList<GladeAtkRelation>, list of atk relations.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the accessibility info to a formatted String.
  - **Returns** - String, formatted String with the accessibility's data.

1.1.3 CLASS **GladeAtkAction**

ValueObject with the data of a GladeAtkAction item. <br/>Responsibilities:

- Store the gladeAtkAction data.</li>

DECLARATION

public class GladeAtkAction
extends java.lang.Object

CONSTRUCTORS

- *GladeAtkAction*
  `public` **GladeAtkAction**`( )`
  - **Usage**
    - Default GladeAtkAction constructor.

- *GladeAtkAction*
  `public` **GladeAtkAction**`( java.lang.String` **action_name**`, java.lang.String` **description** `)`
  - **Usage**
    - GladeAtkAction constructor that initializes atk action values.
  - **Parameters**
    - `action_name` - String, atk action name.
    - `description` - String, atk action description.

METHODS

- *getActionName*
  `public String` **getActionName**`( )`
  - **Usage**
    - Gets the atk action name.
  - **Returns** - String, atk action name.

- *getDescription*
  `public String` **getDescription**`( )`
  - **Usage**
    - Gets the atk action description.
  - **Returns** - String, atk action description.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *setActionName*
  `public void` **setActionName**`( java.lang.String` **action_name** `)`

  - Sets up the atk action name.
- **Parameters**
  - `action_name` - String, atk action name.

- *setDescription*
  `public void` **setDescription**`( java.lang.String` **description** `)`
  - **Usage**
    - Sets up the atk action description.
  - **Parameters**
    - `description` - String, atk action description.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the atk action info to a formatted String.
  - **Returns** - String, formatted String with the atk action's data.

1.1.4 Class **GladeAtkRelation**

ValueObject with the data of a GladeAtkRelation item. <br/>Responsibilities:

- Store the gladeAtkRelation data.</li>

<br/>

Declaration

public class GladeAtkRelation
extends java.lang.Object

Constructors

- *GladeAtkRelation*
  `public` **GladeAtkRelation**`( )`
  - **Usage**
    - Default GladeAtkRelation constructor.

- *GladeAtkRelation*
  `public` **GladeAtkRelation**`( java.lang.String` **target**`, java.lang.String` **type** `)`
  - **Usage**
    - GladeAtkRelation constructor that initializes atk relation values.
  - **Parameters**
    - `target` - String, atk relation target.

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getTarget*
  `public String` **getTarget**`( )`
  - **Usage**
    - Gets atk relation target.
  - **Returns** - String, atk relation target.

- *getType*
  `public String` **getType**`( )`
  - **Usage**
    - Gets atk relation type.
  - **Returns** - String, atk relation type.

- *setTarget*
  `public void` **setTarget**`( java.lang.String` **target** `)`
  - **Usage**
    - Sets up atk relation target.
  - **Parameters**
    - `target` - String, atk relation target.

- *setType*
  `public void` **setType**`( java.lang.String` **type** `)`
  - **Usage**
    - Sets up atk relation type.
  - **Parameters**
    - `type` - String, atk relation type.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the atk relation info to a formatted String.
  - **Returns** - String, formatted String with the atk relation's data.

1.1.5 CLASS **GladeChild**

ValueObject with the data of a GladeChild item. <br/>Responsibilities:

- Store the gladeChild data.</li>

DECLARATION

public class GladeChild
extends java.lang.Object

CONSTRUCTORS

- *GladeChild*
  `public` **GladeChild**`( )`
  - **Usage**
    - Default GladeChild constructor.

- *GladeChild*
  `public` **GladeChild**`( java.lang.String` **internalChild** `)`
  - **Usage**
    - GladeChild constructor that initializes child values.
  - **Parameters**
    - `internalChild` - String.

- *GladeChild*
  `public` **GladeChild**`( java.lang.String` **internalChild**`,`
  `com.imagos.javaglade.vo.GladePlaceHolder` **placeHolder**`,`
  `com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - GladeChild constructor that initializes child values. It creates a child of type PLACEHOLDER, that is, a blank space where widgets can be inserted.
  - **Parameters**
    - `internalChild` - String.
    - `placeHolder` - GladePlaceHolder, Child place holder data.
    - `packing` - GladePacking, packing properties to place the child widget.

- *GladeChild*
  `public` **GladeChild**`( java.lang.String` **internalChild**`,`
  `com.imagos.javaglade.vo.GladeWidget` **widget**`,`
  `com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - GladeChild constructor that initializes child values. It creates a child of type WIDGET.
  - **Parameters**
    - `internalChild` - String.
    - `widget` - GladeWidget, Child widget data.
    - `packing` - GladePacking, packing properties to place the child widget.

METHODS

- *getInternalChild*
  `public String` **getInternalChild**`( )`
  - **Usage**
    - Gets the value of internal child.
  - **Returns** - String, internal child.

- *getPacking*
  `public GladePacking` **getPacking**`( )`
  - **Usage**
    - Gets the child packing information.
  - **Returns** - GladePacking, packing data for widget placement into the parent.

- *getPlaceHolder*
  `public GladePlaceHolder` **getPlaceHolder**`( )`
  - **Usage**
    - Gets the place holder of the child (NULL if the child is not of type PLACEHOLDER).
  - **Returns** - GladePlaceHolder, child placeholder data.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getWidget*
  `public GladeWidget` **getWidget**`( )`
  - **Usage**
    - Gets the widget of the child (NULL if the child is not of type WIDGET).
  - **Returns** - GladeWidget, child widget data.

- *setInternalChild*
  `public void` **setInternalChild**`( java.lang.String` **internalChild** `)`
  - **Usage**
    - Sets up the value of internal child.
  - **Parameters**
    - `internalChild` - String, internal child.

- *setPacking*
  `public void` **setPacking**`( com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Sets up the child packing information.

    - `packing` - GladePacking, packing data for widget placement into the parent.

- *setPlaceHolder*
  `public void` **setPlaceHolder**`( com.imagos.javaglade.vo.GladePlaceHolder` **placeHolder** `)`
  - **Usage**
    - Sets up the place holder of the child.
  - **Parameters**
    - `placeHolder` - GladePlaceHolder, child placeholder data.

- *setWidget*
  `public void` **setWidget**`( com.imagos.javaglade.vo.GladeWidget` **widget** `)`
  - **Usage**
    - Sets up the widget of the child.
  - **Parameters**
    - `widget` - GladeWidget, child widget data.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the child info to a formatted String.
  - **Returns** - String, formatted String with the child's data.

1.1.6 CLASS **GladeInterface**

ValueObject with the data of a parsed Glade Interface. <br/>Responsibilities:

- Store the glade interface data.</li>

<br/>

DECLARATION

public class GladeInterface
extends java.lang.Object

CONSTRUCTORS

- *GladeInterface*
  `public` **GladeInterface**`( )`
  - **Usage**
    - Default GladeInterface constructor.

- *GladeInterface*

- **Usage**
  - GladeInterface constructor that initializes the interface values.
- **Parameters**
  - `xmlns` - String, path to the glade validation DTD file.

Methods

- *addName*
  `public void` **addName**`( java.lang.String` **widgetName**`,`
  `com.imagos.javaglade.vo.GladeWidget` **widget** `)`
  - **Usage**
    - Adds a new widget to the interface catalog of widgets.
  - **Parameters**
    - `widgetName` - String, widget's name.
    - `widget` - GladeWidget, widget's data to insert.

- *addWidget*
  `public void` **addWidget**`( com.imagos.javaglade.vo.GladeWidget` **widget** `)`
  - **Usage**
    - Adds a new widget to the interface toplevel widgets' list.
  - **Parameters**
    - `widget` - GladeWidget, widget's data to insert.

- *getNames*
  `public Hashtable` **getNames**`( )`
  - **Usage**
    - Gets the catalog of widgets created from the glade file.
  - **Returns** - Hashtable<String, GladeWidget>, table of created widgets, indexed by name.

- *getProject*
  `public GladeProject` **getProject**`( )`
  - **Usage**
    - Gets the glade project data.
  - **Returns** - GladeProject, glade project data.

- *getRequires*
  `public ArrayList` **getRequires**`( )`
  - **Usage**
    - Gets the list of required libs created from the glade file.
  - **Returns** - ArrayList<String>, list of required libs.

- *getRevision*

  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getToplevels*
  `public ArrayList` **getToplevels**`( )`
  - **Usage**
    - Gets the list of toplevel widgets (windows and dialogs) created from the glade file.
  - **Returns** - ArrayList<GladeWidget>, list of toplevel widgets.

- *getXmlns*
  `public String` **getXmlns**`( )`
  - **Usage**
    - Gets the path to the glade DTD validation file.
  - **Returns** - String, glade DTD validation file.

- *setNames*
  `public void` **setNames**`( java.util.Hashtable` **names** `)`
  - **Usage**
    - Sets up the catalog of widgets created from the glade file.
  - **Parameters**
    - `names` - Hashtable<String, GladeWidget>, table of created widgets, indexed by name.

- *setProject*
  `public void` **setProject**`( com.imagos.javaglade.vo.GladeProject` **project** `)`
  - **Usage**
    - Sets up the glade project data.
  - **Parameters**
    - `project` - GladeProject, glade project data.

- *setRequires*
  `public void` **setRequires**`( java.util.ArrayList` **requires** `)`
  - **Usage**
    - Sets up the list of required libs created from the glade file.
  - **Parameters**
    - `requires` - ArrayList<String>, list of required libs.

- *setToplevels*
  `public void` **setToplevels**`( java.util.ArrayList` **toplevels** `)`
  - **Usage**
    - Sets up the list of toplevel widgets (windows and dialogs) created from the glade file.
  - **Parameters**
    - `toplevels` - ArrayList<GladeWidget>, list of toplevel widgets.

- *setXmlns*
  `public void` **setXmlns**`( java.lang.String` **xmlns** `)`
  - **Usage**
    - Sets up the glade project data.
  - **Parameters**
    - `xmlns` - String, glade DTD validation file.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the glade interface info to a formatted String.
  - **Returns** - String, formatted String with the interface's data.

1.1.7 CLASS GladePacking

ValueObject with the data of a GladePacking item. <br/>Responsibilities:

- Store the gladePacking data.</li>

<br/>

DECLARATION

public class GladePacking
extends java.lang.Object

CONSTRUCTORS

- *GladePacking*
  `public` **GladePacking**`( )`

METHODS

- *addProperty*
  `public void` **addProperty**`( com.imagos.javaglade.vo.GladeProperty` **prop** `)`
  - **Usage**
    - Adds a new property to the packing properties' list.
  - **Parameters**
    - `prop` - GladeProperty, packing property's data to insert.

- *getLProperties*
  `public ArrayList` **getLProperties**`( )`
  - **Usage**

  - **Returns** - ArrayList<GladeProperty>, list of packing properties.

- *getProperty*
  `public GladeProperty` **getProperty**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets a property contained in the packing properties' list.
  - **Parameters**
    - `name` - String, packing property's name.
  - **Returns** - GladeProperty, found packing property's data.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *setLProperties*
  `public void` **setLProperties**`( java.util.ArrayList` **properties** `)`
  - **Usage**
    - Sets up the list of packing properties.
  - **Parameters**
    - `properties` - ArrayList<GladeProperty>, list of packing properties.

- *setProperty*
  `public void` **setProperty**`( com.imagos.javaglade.vo.GladeProperty` **prop** `)`
  - **Usage**
    - Updates the value of a property contained in the packing properties' list.
  - **Parameters**
    - `prop` - GladeProperty, packing property's new data.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the packing info to a formatted String.
  - **Returns** - String, formatted String with the packing's data.

1.1.8 CLASS **GladePlaceHolder**

ValueObject with the data of a GladePlaceHolder item. <br/>Responsibilities:

- Store the gladePlaceHolder data.</li>

<br/>

DECLARATION

public class GladePlaceHolder
extends java.lang.Object

CONSTRUCTORS

- *GladePlaceHolder*
  `public` **GladePlaceHolder**`( )`

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the placeholder info to a formatted String.
  - **Returns** - String, formatted String with the placeholder's data.

1.1.9 CLASS **GladeProject**

ValueObject with the data of a GladeProject item. <br/>Responsibilities:

- Store the gladeProject data.</li>

<br/>

DECLARATION

public class GladeProject
extends java.lang.Object

CONSTRUCTORS

- *GladeProject*
  `public` **GladeProject**`( )`

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the project info to a formatted String.
  - **Returns** - String, formatted String with the project's data.

1.1.10 CLASS **GladeProperty**

ValueObject with the data of a GladeProperty item. <br/>Responsibilities:

- Store the gladeProperty data.</li>

<br/>

DECLARATION

public class GladeProperty
extends java.lang.Object

CONSTRUCTORS

- *GladeProperty*
  `public` **GladeProperty**`( )`
  - **Usage**
    - Default GladeProperty constructor.

- *GladeProperty*
  `public` **GladeProperty**`( java.lang.String` **name**`, java.lang.String` **value**`, boolean` **translatable** `)`
  - **Usage**
    - GladeProperty constructor that initializes property values.
  - **Parameters**
    - `name` - String, name of the property.
    - `value` - String, value of the property casted as String.
    - `translatable` - boolean, True if the property value should be translated into the user's locale language; False in any other case.

- *GladeProperty*
  `public` **GladeProperty**`( java.lang.String` **name**`, java.lang.String` **value**`, java.lang.String` **translatable** `)`
  - **Usage**
    - GladeProperty constructor that initializes property values.
  - **Parameters**
    - `name` - String, name of the property.

    - `translatable` - String, Boolean value in String format. True if the property value should be translated into the user's locale language; False in any other case.

- *GladeProperty*
  `public` **GladeProperty**`( java.lang.String` **name**`, java.lang.String` **value**`, java.lang.String` **translatable**`, java.lang.String` **type**`, java.lang.String` **context**`, java.lang.String` **comments**`, java.lang.String` **agent** `)`
  - **Usage**
    - GladeProperty constructor that initializes property values.
  - **Parameters**
    - `name` - String, name of the property.
    - `value` - String, value of the property casted as String.
    - `translatable` - boolean, True if the property value should be translated into the user's locale language; False in any other case.
    - `type` - String, type of the property.
    - `context` - String, context in which the property is defined.
    - `comments` - String, comments linked to the property.
    - `agent` - String, property's agent.

METHODS

- *equals*
  `public boolean` **equals**`( java.lang.Object` **obj** `)`
  - **Usage**
    - Compares two properties to determine if both of them are equal. Two properties will be equal if their name is the same.
  - **Parameters**
    - `obj` - Object, property info to compare to.

- *getAgent*
  `public String` **getAgent**`( )`
  - **Usage**
    - Gets the property agent.
  - **Returns** - String, property agent.

- *getComments*
  `public String` **getComments**`( )`
  - **Usage**
    - Gets the property comments.
  - **Returns** - String, property comments.

- *getContext*
  `public String` **getContext**`( )`
  - **Usage**
    - Gets the property context.

- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Gets the property name.
  - **Returns** - String, property name.
- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.
- *getType*
  `public String` **getType**`( )`
  - **Usage**
    - Gets the property type.
  - **Returns** - String, property type.
- *getValue*
  `public String` **getValue**`( )`
  - **Usage**
    - Gets the property value formatted as a String.
  - **Returns** - String, property value in String format.
- *isTranslatable*
  `public boolean` **isTranslatable**`( )`
  - **Usage**
    - Gets the value of the translatable flag for the property.
  - **Returns** - boolean, True if the property should be translated into the user's locale language; False in any other case.
- *setAgent*
  `public void` **setAgent**`( java.lang.String` **agent** `)`
  - **Usage**
    - Sets up the property agent.
  - **Parameters**
    - `agent` - String, property agent.
- *setComments*
  `public void` **setComments**`( java.lang.String` **comments** `)`
  - **Usage**
    - Sets up the property comments.
  - **Parameters**
    - `comments` - String, property comments.

- *setContext*
  `public void` **setContext**`( java.lang.String` **context** `)`
  - **Usage**
    - Sets up the property context.
  - **Parameters**
    - `context` - String, property context.

- *setI18n*
  `public void` **setI18n**`( I18n` **i18n** `)`
  - **Usage**
    - Sets up the instance of Internalization object used for the property value localization.
  - **Parameters**
    - `i18n` - I18n, Internationalization instance.

- *setName*
  `public void` **setName**`( java.lang.String` **name** `)`
  - **Usage**
    - Sets up the property name.
  - **Parameters**
    - `name` - String, property name.

- *setTranslatable*
  `public void` **setTranslatable**`( boolean` **translatable** `)`
  - **Usage**
    - Sets up the value of the translatable flag for the property.
  - **Parameters**
    - `translatable` - boolean, value of the translatable flag.

- *setType*
  `public void` **setType**`( java.lang.String` **type** `)`
  - **Usage**
    - Sets up the property type.
  - **Parameters**
    - `type` - String, property type.

- *setValue*
  `public void` **setValue**`( java.lang.String` **value** `)`
  - **Usage**
    - Sets up the property value.
  - **Parameters**
    - `value` - String, property value.

- *toString*

- **Usage**
  - Converts the property info to a formatted String.
- **Returns** - String, formatted String with the property's data.

1.1.11 Class **GladeSignal**

ValueObject with the data of a GladeSignal item. <br/>Responsibilities:

- Store the gladeSignal data.</li>

<br/>

DECLARATION

---

public class GladeSignal
extends java.lang.Object

CONSTRUCTORS

- *GladeSignal*
  `public` **GladeSignal**`( )`
  - **Usage**
    - Default GladeSignal constructor

- *GladeSignal*
  `public` **GladeSignal**`( java.lang.String` **name**`, java.lang.String` **handler**`, java.util.Date` **lastModification** `)`
  - **Usage**
    - GladeSignal constructor that initializes signal values.
  - **Parameters**
    - `name` - String, name of the handled signal.
    - `handler` - String, name of the handler method.
    - `lastModification` - Date, date of last modification of the signal in the glade file.

- *GladeSignal*
  `public` **GladeSignal**`( java.lang.String` **name**`, java.lang.String` **handler**`, java.lang.String` **lastModification** `)`
  - **Usage**
    - GladeSignal constructor that initializes signal values.
  - **Parameters**
    - `name` - String, name of the handled signal.
    - `handler` - String, name of the handler method.
    - `lastModification` - String, date of last modification of the signal in the glade file (in string format).

- *GladeSignal*
  `public` **GladeSignal**`( java.lang.String` **name**`, java.lang.String` **handler**`, java.lang.String` **object**`, boolean` **after**`, java.util.Date` **lastModification** `)`
  - **Usage**
    - GladeSignal constructor that initializes signal values.
  - **Parameters**
    - `name` - String, name of the handled signal.
    - `handler` - String, name of the handler method.
    - `object` - String, classpath of the class that implements the signal handler.
    - `after` - boolean.
    - `lastModification` - Date, date of last modification of the signal in the glade file.

- *GladeSignal*
  `public` **GladeSignal**`( java.lang.String` **name**`, java.lang.String` **handler**`, java.lang.String` **object**`, java.lang.String` **after**`, java.lang.String` **lastModification** `)`
  - **Usage**
    - GladeSignal constructor that initializes signal values.
  - **Parameters**
    - `name` - String, name of the handled signal.
    - `handler` - String, name of the handler method.
    - `object` - String, classpath of the class that implements the signal handler.
    - `after` - String.
    - `lastModification` - String, date of last modification of the signal in the glade file.

METHODS

- *getHandler*
  `public String` **getHandler**`( )`
  - **Usage**
    - Gets the signal handler method name.
  - **Returns** - String, signal handler method.

- *getLastModification*
  `public Date` **getLastModification**`( )`
  - **Usage**
    - Gets the signal last modification date.
  - **Returns** - Date, signal last modification date.

- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Gets the signal name.
  - **Returns** - String, signal name.

- *getObject*
  `public String` **getObject**`( )`
  - **Usage**
    - Gets the classpath of the handler class.
  - **Returns** - String, signal handler class.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *isAfter*
  `public boolean` **isAfter**`( )`
  - **Usage**
    - Gets the value of after for the signal.
  - **Returns** - boolean.

- *setAfter*
  `public void` **setAfter**`( boolean` **after** `)`
  - **Usage**
    - Sets up the value of after for the signal.
  - **Parameters**
    - `after` - boolean.

- *setHandler*
  `public void` **setHandler**`( java.lang.String` **handler** `)`
  - **Usage**
    - Sets up the signal handler method name.
  - **Parameters**
    - `handler` - String, signal handler method.

- *setLastModification*
  `public void` **setLastModification**`( java.util.Date` **lastModification** `)`
  - **Usage**
    - Sets up the signal last modification date.
  - **Parameters**
    - `lastModification` - Date, signal last modification date.

- *setName*
  `public void` **setName**`( java.lang.String` **name** `)`
  - **Usage**
    - Sets up the signal name.
  - **Parameters**

- *setObject*
  `public void setObject( java.lang.String object )`
  - **Usage**
    - Sets up the classpath of the handler class.
  - **Parameters**
    - `object` - String, signal handler class.

- *toString*
  `public String toString( )`
  - **Usage**
    - Converts the signal info to a formatted String.
  - **Returns** - String, formatted String with the signal's data.

1.1.12 CLASS GladeStyle

ValueObject with the data of a GladeStyle item. <br/>Responsibilities:

- Store the gladeStyle data.</li>

<br/>

DECLARATION

public class GladeStyle
extends java.lang.Object

CONSTRUCTORS

- *GladeStyle*
  `public GladeStyle( )`

METHODS

- *getRevision*
  `public static String getRevision( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *toString*
  `public String toString( )`
  - **Usage**
    - Converts the style info to a formatted String.

1.1.13 Class **GladeWidget**

ValueObject with the data of a GladeWidget item. <br/>Responsibilities:

- Store the gladeWidget data.</li>

<br/>

Declaration

public class GladeWidget
extends java.lang.Object

Constructors

- *GladeWidget*
  `public` **GladeWidget**`( )`
  - **Usage**
    - Default GladeWidget constructor.

- *GladeWidget*
  `public` **GladeWidget**`( java.lang.String` **classname**, `java.lang.String` **id** `)`
  - **Usage**
    - GladeWidget constructor that initializes widget values.
  - **Parameters**
    - `classname` - String, Widget's class (type).
    - `id` - String, Widget's unique identifier.

Methods

- *addAccelerator*
  `public void` **addAccelerator**`( com.imagos.javaglade.vo.GladeAccelerator` **accel** `)`
  - **Usage**
    - Adds a new accelerator to the widget accelerator' list.
  - **Parameters**
    - `accel` - GladeAccelerator, accelerator's data to insert.

- *addChild*
  `public void` **addChild**`( com.imagos.javaglade.vo.GladeChild` **child** `)`
  - **Usage**
    - Adds a new child widget to the widget children' list.
  - **Parameters**

- *addProperty*
  `public void` **addProperty**`( com.imagos.javaglade.vo.GladeProperty` **prop** `)`
  - **Usage**
    - Adds a new property to the widget properties' list.
  - **Parameters**
    - `prop` - GladeProperty, property's data to insert.

- *addSignal*
  `public void` **addSignal**`( com.imagos.javaglade.vo.GladeSignal` **signal** `)`
  - **Usage**
    - Adds a new signal to the widget signals' list.
  - **Parameters**
    - `signal` - GladeSignal, signal's data to insert.

- *getAccessibility*
  `public GladeAccessibility` **getAccessibility**`( )`
  - **Usage**
    - Gets the widget accesibility data.
  - **Returns** - GladeAccessibility, widget accesibility data.

- *getClassname*
  `public String` **getClassname**`( )`
  - **Usage**
    - Gets the widget classname (type).
  - **Returns** - String, widget classname.

- *getId*
  `public String` **getId**`( )`
  - **Usage**
    - Gets the widget unique identifier.
  - **Returns** - String, widget identifier.

- *getLAccelators*
  `public ArrayList` **getLAccelators**`( )`
  - **Usage**
    - Gets the list of accelerator defined for the widget.
  - **Returns** - ArrayList<GladeAccelerator>, list of accelerators.

- *getLChildren*
  `public ArrayList` **getLChildren**`( )`
  - **Usage**
    - Gets the list of children widgets defined for the widget.

- *getLProperties*
  `public ArrayList` **getLProperties**`( )`
  - **Usage**
    - Gets the list of widget properties.
  - **Returns** - ArrayList<GladeProperty>, list of widget's properties.

- *getLSignals*
  `public ArrayList` **getLSignals**`( )`
  - **Usage**
    - Gets the list of handled signals for the widget.
  - **Returns** - ArrayList<GladeSignal>, list of widget's signals.

- *getParent*
  `public GladeWidget` **getParent**`( )`
  - **Usage**
    - Gets the parent widget.
  - **Returns** - GladeWidget, parent widget.

- *getProperty*
  `public String` **getProperty**`( java.lang.String` **propName** `)`
  - **Usage**
    - Gets the value of a widget's property.
  - **Parameters**
    - `propName` - String, name of the searched property.
  - **Returns** - String, value of the widget's property in String format.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *setAccessibility*
  `public void` **setAccessibility**`( com.imagos.javaglade.vo.GladeAccessibility` **accessibility** `)`
  - **Usage**
    - Sets up the widget accesibility data.
  - **Parameters**
    - `accessibility` - GladeAccessibility, widget accesibility data.

- *setClassname*
  `public void` **setClassname**`( java.lang.String` **classname** `)`
  - **Usage**

  - **Parameters**
    - `classname` - String, widget classname.

- *setId*
  `public void` **setId**`( java.lang.String` **id** `)`
  - **Usage**
    - Sets up the widget unique identifier.
  - **Parameters**
    - `id` - String, widget identifier.

- *setLAccelators*
  `public void` **setLAccelators**`( java.util.ArrayList` **accelerators** `)`
  - **Usage**
    - Sets up the list of accelerator defined for the widget.
  - **Parameters**
    - `accelerators` - ArrayList<GladeAccelerator>, list of accelerators.

- *setLChildren*
  `public void` **setLChildren**`( java.util.ArrayList` **children** `)`
  - **Usage**
    - Sets up the list of children widgets defined for the widget.
  - **Parameters**
    - `children` - ArrayList<GladeChild>, list of widget's children.

- *setLProperties*
  `public void` **setLProperties**`( java.util.ArrayList` **properties** `)`
  - **Usage**
    - Sets up the list of widget properties.
  - **Parameters**
    - `properties` - ArrayList<GladeProperty>, list of widget's properties.

- *setLSignals*
  `public void` **setLSignals**`( java.util.ArrayList` **signals** `)`
  - **Usage**
    - Sets up the list of handled signals for the widget.
  - **Parameters**
    - `signals` - ArrayList<GladeSignal>, list of widget's signals.

- *setParent*
  `public void` **setParent**`( com.imagos.javaglade.vo.GladeWidget` **parent** `)`
  - **Usage**
    - Sets up the parent widget.
  - **Parameters**
    - `parent` - GladeWidget, parent widget

- *toString*
  `public String` **`toString`**`( )`
  - **Usage**
    - Converts the widget info to a formatted String.
  - **Returns** - String, formatted String with the widget's data.

Chapter 2

Package com.imagos.javaglade

Package Contents *Page*

Interfaces

InterfaceGladeParser 33
Interface with the constant definition of the values used in the parsing of the glade files.

Localizer 39
...no description...

Classes

GenericListener 40
Implementation of a Generic Event Listener that can handle any kind of event produced.

Gui 41
Implements an user GUI.

InterfaceGladeGtk 45
Interface with the constant definition of the values used in the Glade for Java module.

JavaGlade 71
Main class of the JavaGlade module.

JavaGladeCatalog 73
Implements the catalog of widgets built by the JavaGlade conversion module.

JavaGladeException 75
This class encapsulates and holds all the errors that might have taken place during any of the Glade to Java conversion operations.

JavaGladeTranslator 77
This class functions as a Factory for I18n instances that will be used to translate the texts into the user's current locale.

JavaGladeUtil 78
Faade of the JavaGlade module.

2.1 Interfaces

2.1.1 INTERFACE **InterfaceGladeParser**

Interface with the constant definition of the values used in the parsing of the glade files. It holds glade tagnames, glade attribute names, etc. <br/>

DECLARATION

```
public interface InterfaceGladeParser
```

FIELDS

- public static final String GLADE_INTERFACE_TAG
  - Name of the glade-interface tag in the glade file
- public static final String PROJECT_TAG
  - Name of the project tag in the glade file
- public static final String REQUIRES_TAG
  - Name of the requires tag in the glade file
- public static final String STYLE_TAG
  - Name of the style tag in the glade file
- public static final String WIDGET_TAG
  - Name of the widget tag in the glade file
- public static final String PROPERTY_TAG
  - Name of the property tag in the glade file
- public static final String ACCESSIBILITY_TAG
  - Name of the accessibility tag in the glade file
- public static final String SIGNAL_TAG
  - Name of the signal tag in the glade file
- public static final String ACCELERATOR_TAG
  - Name of the accelerator tag in the glade file
- public static final String CHILD_TAG
  - Name of the child tag in the glade file
- public static final String ATK_RELATION_TAG
  - Name of the atk-relation tag in the glade file

  - Name of the atk-action tag in the glade file

- public static final String ATK_PROPERTY_TAG
  - Name of the atk-property tag in the glade file
- public static final String PLACEHOLDER_TAG
  - Name of the placeholder tag in the glade file
- public static final String PACKING_TAG
  - Name of the packing tag in the glade file
- public static final String GLADEINTERFACE_XMLNS_PROPERTYNAME
  - Name of the xmlns property of the item <glade-interface>
- public static final String PROPERTY_NAME_PROPERTYNAME
  - Name of the name property of the item <property>
- public static final String PROPERTY_VALUE_PROPERTYNAME
  - Name of the value property of the item <property>
- public static final String PROPERTY_TYPE_PROPERTYNAME
  - Name of the type property of the item <property>
- public static final String PROPERTY_TRANSLATABLE_PROPERTYNAME
  - Name of the translatable property of the item <property>
- public static final String PROPERTY_CONTEXT_PROPERTYNAME
  - Name of the context property of the item <property>
- public static final String PROPERTY_COMMENTS_PROPERTYNAME
  - Name of the comments property of the item <property>
- public static final String PROPERTY_AGENT_PROPERTYNAME
  - Name of the agent property of the item <property>
- public static final String SIGNAL_NAME_PROPERTYNAME
  - Name of the name property of the item <signal>
- public static final String SIGNAL_HANDLER_PROPERTYNAME
  - Name of the handler property of the item <signal>
- public static final String SIGNAL_LASTMODIFICATION_PROPERTYNAME
  - Name of the lastModification property of the item <signal>
- public static final String SIGNAL_AFTER_PROPERTYNAME
  - Name of the after property of the item <signal>
- public static final String SIGNAL_OBJECT_PROPERTYNAME
  - Name of the object property of the item <signal>

  - Name of the key property of the item
- public static final String ACCELERATOR_SIGNAL_PROPERTYNAME
  - Name of the signal property of the item
- public static final String ACCELERATOR_MODIFIERS_PROPERTYNAME
  - Name of the modifiers property of the item
- public static final String CHILD_INTERNAL_PROPERTYNAME
  - Name of the inline property of the item <child>
- public static final String WIDGET_CLASS_PROPERTYNAME
  - Name of the class property of the item <widget>
- public static final String WIDGET_ID_PROPERTYNAME
  - Name of the id property of the item <widget>
- public static final String REQUIRES_LIB_PROPERTYNAME
  - Name of the lib property of the item <requires>
- public static final String ATKPROPERTY_NAME_PROPERTYNAME
  - Name of the name property of the item
- public static final String ATKPROPERTY_VALUE_PROPERTYNAME
  - Name of the value property of the item
- public static final String ATKPROPERTY_TYPE_PROPERTYNAME
  - Name of the type property of the item
- public static final String ATKPROPERTY_TRANSLATABLE_PROPERTYNAME
  - Name of the translatable property of the item
- public static final String ATKPROPERTY_CONTEXT_PROPERTYNAME
  - Name of the context property of the item
- public static final String ATKPROPERTY_COMMENTS_PROPERTYNAME
  - Name of the comments property of the item
- public static final String ATKRELATION_TARGET_PROPERTYNAME
  - Name of the target property of the item
- public static final String ATKRELATION_TYPE_PROPERTYNAME
  - Name of the type property of the item
- public static final String ATKACTION_ACTIONNAME_PROPERTYNAME
  - Name of the action_name property of the item
- public static final String ATKACTION_DESCRIPTION_PROPERTYNAME
  - Name of the description property of the item

  –

- public static final int ERROR_UNKNOWN_TAG
  –
- public static final int ERROR_UNEXPECTED_TAG
  –
- public static final int ERROR_GLADE_PARSER
  –
- public static final String JAVAGLADE_LOG_CATEGORY
  - Category of the logging system for the JavaGlade module.
- public static final String JAVAGLADE_LOG_AUTOMATICLISTENER
  - Message for logging the implementation of an automatic listener for a textArea widget.
- public static final String JAVAGLADE_LOG_NEWWIDGET
  - Message for logging the building of a new widget.
- public static final String JAVAGLADE_LOG_NEWSIGNAL
  - Message for logging the building of a new signal.
- public static final String JAVAGLADE_LOG_MENU_WITH_NO_WINDOW
  - Message for logging an error when a menu is created and there is no toplevel window.
- public static final String JAVAGLADE_LOG_MENUITEM_WITH_NO_MENU
  - Message for logging an error when a menuitem is created and its parent is not a menu widget.
- public static final String JAVAGLADE_LOG_TOOLITEM_WITH_NO_TOOLBAR
  - Message for logging an error when a toolitem is created and its parent is not a toolbar widget.
- public static final String JAVAGLADE_LOG_COOLITEM_WITH_NO_COOLBAR
  - Message for logging an error when a coolitem is created and its parent is not a coolbar widget.
- public static final String JAVAGLADE_LOG_BUILD_TOOLITEM_ERROR
  - Message for logging error during the construction of the toolItem widget.
- public static final String JAVAGLADE_LOG_COMPOSITE_HOMOGENEOUS_ERROR
  - Message for logging an error produced while checking if a composite is homogeneous.
- public static final String JAVAGLADE_LOG_CONSTRUCT
  - Message for logging JavaGlade construction
- public static final String JAVAGLADE_LOG_CREATE_TOOLKIT
  - Message for logging SWT Toolkit creation.

  - Message for logging the building of the JavaGlade interface.

- public static final String JAVAGLADE_LOG_BUILD_INTERFACE_COMPLETED
  - Message for logging the end of the building of the JavaGlade interface.
- public static final String JAVAGLADE_LOG_BUILD_WIDGET
  - Message for logging the building of a new widget.
- public static final String JAVAGLADE_LOG_BUILD_WIDGET_SEPARATOR
  - Message for separate the logging of different widgets.
- public static final String JAVAGLADE_LOG_BUILD_WIDGET_PROPERTIES
  - Message for logging the properties of a built widget.
- public static final String JAVAGLADE_LOG_BUILD_WIDGET_PACKING
  - Message for logging the packing data of a built widget.
- public static final String JAVAGLADE_LOG_AVOID_NON_COMPOSITE_PARENT
  - Message for logging the non-composite parent incompatibility. In GTK some widgets can have children even though they are not of type composite (for example Button). In SWT only composite widgets can have children.
- public static final String JAVAGLADE_LOG_NULL_PARENT
  - Message for logging the null-parent error.
- public static final String JAVAGLADE_LOG_CREATE_WIDGET
  - Message for logging the beginning of the building of a new widget.
- public static final String JAVAGLADE_LOG_BUILD_RADIO_GROUP
  - Message for logging the building of a radiogroup widget.
- public static final String JAVAGLADE_LOG_NULL_RADIO_GROUP
  - Message for logging an error in the building of a radiogroup widget.
- public static final String JAVAGLADE_LOG_IGNORE_GTKALIGNMENT
  - Message for logging the ignoring of GtkAlignment widgets.
- public static final String JAVAGLADE_LOG_UNSUPPORTED_WIDGET
  - Message for logging unsupported widget class.
- public static final String JAVAGLADE_LOG_NULL_WIDGET
  - Message for logging error during the construction of the widget (null widget).
- public static final String JAVAGLADE_LOG_ITERATE_CHILDREN
  - Message for logging the beginning of children iteration.
- public static final String JAVAGLADE_LOG_CHILDREN_WITH_NO_WIDGET
  - Message for logging the beginning of children iteration.

  - Message for logging the building of a popup menu not linked to a menuItem.

- public static final String JAVAGLADE_LOG_MENUITEM_WITH_NO_MENUBAR
  - Message for logging the building of a menuItem not linked to a menuBar.
- public static final String JAVAGLADE_LOG_NOTEBOOK_WITH_NO_PARENT
  - Message for logging the building of a notebook with no parent.
- public static final String JAVAGLADE_LOG_NULL_TABITEM
  - Message for logging error during the construction of the tabItem widget (null widget).
- public static final String JAVAGLADE_LOG_NULL_BUTTON
  - Message for logging error during the construction of the button widget (null widget).
- public static final String JAVAGLADE_LOG_BUILD_BUTTON_ERROR
  - Message for logging error during the construction of the button widget.
- public static final String JAVAGLADE_LOG_MENUBAR_WITH_NO_PARENT
  - Message for logging the building of a menuBar with no parent.
- public static final String JAVAGLADE_LOG_POPUP_WITH_NO_PARENT
  - Message for logging the building of a popup menu with no parent.
- public static final String JAVAGLADE_LOG_CONNECT_SIGNAL
  - Message for logging the creation of a new signal.
- public static final String JAVAGLADE_LOG_OPEN_DIALOG
  - Message for logging the opening of a dialog window.
- public static final String JAVAGLADE_LOG_REQUIRE
  - Message for logging a require element.
- public static final String JAVAGLADE_LOG_SEARCH_IN_GLADE_CHILD
  - Message for logging a widget search in a JavaGlade child.
- public static final String JAVAGLADE_LOG_ADD_GLADE_CHILD
  - Message for logging the adding of a new JavaGlade instance (embedded Glade file).
- public static final String JAVAGLADE_LOG_NO_TOPLEVELS
  - Message for logging an error in printing build widgets information (no widgets built).
- public static final String JAVAGLADE_LOG_SEARCH_NULL_WIDGET_NAME
  - Message for logging an error in searching a widget (no widget name).
- public static final String JAVAGLADE_LOG_CREATE_PROPERTY
  - Message for logging the building of a new Glade property.
- public static final String JAVAGLADE_LOG_TRANSLATE_TEXT
  - Message for logging the translation of a JavaGlade property text/value.

  - Message for logging the error in translation of a JavaGlade property text/value (No internationalization object provided).

- public static final String JAVAGLADE_LOG_RESOLVE_DTD
  - Message for logging the path of the local version of the Glade DTD used instead of the external URI provided in the glade files.
- public static final String JAVAGLADE_LOG_CREATE_PARSER
  - Message for logging the creation of a valid Glade Parser.
- public static final String JAVAGLADE_LOG_PARSE_GLADE
  - Message for logging the path of the glade file to be parsed
- public static final String JAVAGLADE_LOG_ERROR_GETTING_PARSER
  - Message for logging an error during the creation of the Glade Parser instance.
- public static final String JAVAGLADE_LOG_ERROR_READING_FILE
  - Message for logging an error accessing or reading the glade file to be parsed, such as file not found, or file not accesible or file with no needed permissions, etc.
- public static final String JAVAGLADE_LOG_ERROR_PARSING_FILE
  - Message for logging a general error during the parsing of the Glade file.
- public static final String JAVAGLADE_LOG_PARSING_START_DOCUMENT
  - Message for logging the start of the glade document parsing.
- public static final String JAVAGLADE_LOG_PARSING_START_ELEMENT
  - Message for logging the start of a glade element parsing.
- public static final String JAVAGLADE_LOG_PARSING_END_ELEMENT
  - Message for logging the end of a glade element parsing.
- public static final String JAVAGLADE_LOG_LOADMESSAGES
  - Message for logging the loading of the messages resources file.

2.1.2 INTERFACE **Localizer**

DECLARATION

public interface Localizer

METHODS

- *getCompletePath*
  `public URL` **getCompletePath**`( java.lang.String` **path** `)`
- *getInputStream*

2.2 Classes

2.2.1 Class **GenericListener**

Implementation of a Generic Event Listener that can handle any kind of event produced.
<br/>Responsibilities:

- Link a signal with its handler.</li>
- Register the handler to receive event notifications.</li>

<br/>

Declaration

public class GenericListener
extends java.lang.Object

Constructors

- *GenericListener*
  `public` **GenericListener**`( java.lang.String` **handlerName**`, java.lang.Object` **object** `)`
  - **Usage**
    - Default GenericListener constructor.
  - **Parameters**
    - `handlerName` - String, name of the method that handles the event.
    - `object` - Object, instance of the Event Handler class.

Methods

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *handleEvent*
  `public void` **handleEvent**`( Event` **event** `)`
  - **Usage**
    - Generic handler that receives all the event notificatiions. Later on it will be automatically moved on to the specific handler implementation.
  - **Parameters**
    - `event` - Event, event data.

2.2.2 Class **Gui**

Implements an user GUI. It keeps a record of the widgets defined for the GUI and gives access to them. <br/>Responsibilities:

- Provide accessors for the GUI widgets.</li>
- Dispose GUI widgets.</li>
- Implement the GUI hierarchy.</li>

<br/>

Declaration

public class Gui
extends java.lang.Object

Constructors

- *Gui*
  `public` **Gui**`( )`
- *Gui*
  `public` **Gui**`( java.lang.String` **guiName** `)`
- *Gui*
  `public` **Gui**`( java.lang.String` **guiName**, `com.imagos.javaglade.Gui` **parentGui**, `java.lang.String` **parentWidgetName** `)`
- *Gui*
  `public` **Gui**`( java.lang.String` **guiName**, `com.imagos.javaglade.Gui` **parentGui**, `java.lang.String` **parentWidgetName**, `Shell` **parentShell** `)`

Methods

- *addChild*
  `public void` **addChild**`( com.imagos.javaglade.Gui` **child** `)`
- *addDialog*
  `public void` **addDialog**`( java.lang.String` **name**, `com.imagos.javaglade.toolkit.swt.SwtDialog` **dialog** `)`
  - **Usage**
    - Adds a dialog to the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the dialog.
    - `SwtDialog,` - dialog to add to the catalog.
- *addWidget*
  `public void` **addWidget**`( java.lang.String` **name**, `Widget` **widget**,

  - **Usage**
    - Adds a widget to the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the widget.
    - `Widget` - widget, widget to add to the catalog.
    - `parentName` - String, name of the parent widget.

- *addWindow*
  `public void` **addWindow**`( java.lang.String` **name**`,`
  `com.imagos.javaglade.toolkit.swt.SwtTopLevelWindow` **window** `)`
  - **Usage**
    - Adds a window to the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the window.
    - `SwtTopLevelWindow,` - window to add to the catalog.

- *disposeGui*
  `public void` **disposeGui**`( )`
  - **Usage**
    - Removes all the widgets of the GUI.

- *equals*
  `public boolean` **equals**`( java.lang.Object` **obj** `)`

- *eventLoop*
  `public void` **eventLoop**`( )`
  - **Usage**
    - Opens the toplevel windows and starts the event loop to capture user events. It's the main method to start up the interface.

- *getChild*
  `public Gui` **getChild**`( java.lang.String` **childName** `)`

- *getChildren*
  `public ArrayList` **getChildren**`( )`

- *getDialog*
  `public SwtDialog` **getDialog**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets a dialog from the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the dialog.
  - **Returns** - SwtDialog, found dialog, null in case no dialog has been found.

- *getFullWidgetName*
  `public String` **getFullWidgetName**`( java.lang.String` **name** `)`

- *getGuiName*

- *getParentGui*
  `public Gui` **getParentGui**`( )`

- *getParentWidgetName*
  `public String` **getParentWidgetName**`( )`

- *getParentWindow*
  `public SwtTopLevelWindow` **getParentWindow**`( )`

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getRootWidget*
  `public SwtWidget` **getRootWidget**`( )`

- *getSwtWidget*
  `public SwtWidget` **getSwtWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets a widget from the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the widget.
  - **Returns** - SwtWidget, found widget, null in case no widget has been found.

- *getWidget*
  `public Widget` **getWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets a widget from the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the widget.
  - **Returns** - Widget, found widget, null in case no widget has been found.

- *getWidgetTreeAsString*
  `public String` **getWidgetTreeAsString**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget** `)`

- *getWidgetTreeAsString*
  `public String` **getWidgetTreeAsString**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget**`, java.lang.String` **preffix** `)`

- *getWindow*
  `public SwtTopLevelWindow` **getWindow**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets a window from the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the window.

  - **Returns** - SwtTopLevelWindow, found window, null in case no window has been found.

- *hashCode*
  `public int` **hashCode**`( )`

- *printDetailedWidgets*
  `public void` **printDetailedWidgets**`( Logger` **logger** `)`

- *printDetailedWidgetTree*
  `public void` **printDetailedWidgetTree**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget** `)`

- *printDetailedWidgetTree*
  `public void` **printDetailedWidgetTree**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget**`, Logger` **logger** `)`

- *printWidgets*
  `public void` **printWidgets**`( Logger` **logger** `)`

- *printWidgetTree*
  `public void` **printWidgetTree**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget** `)`

- *printWidgetTree*
  `public void` **printWidgetTree**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget**`, Logger` **logger** `)`

- *removeChild*
  `public void` **removeChild**`( com.imagos.javaglade.Gui` **child** `)`

- *removeWidget*
  `public void` **removeWidget**`( java.lang.String` **name** `)`

- *renameWidget*
  `public void` **renameWidget**`( java.lang.String` **newName**`, java.lang.String` **oldName** `)`
  - **Usage**
    - Renames an existing widget and updates the widgets' catalog.
  - **Parameters**
    - `newName` - String, new name of the widget.
    - `oldName` - String, old name of the widget.

- *setGuiName*
  `public void` **setGuiName**`( java.lang.String` **guiName** `)`

- *setParentGui*
  `public void` **setParentGui**`( com.imagos.javaglade.Gui` **parentGui** `)`

- *setParentWidgetName*
  `public void` **setParentWidgetName**`( java.lang.String` **parentWidgetName** `)`

- *setParentWindow*
  `public void` **setParentWindow**`(`

- *setRootWidget*
  `public void` **setRootWidget**`( java.lang.String` **name** `)`
- *setRootWidget*
  `public void` **setRootWidget**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **rootWidget** `)`

2.2.3 CLASS **InterfaceGladeGtk**

Interface with the constant definition of the values used in the Glade for Java module. It holds Gtk widgets types, Gtk constants, Gtk signals, etc. <br/>

DECLARATION

public class InterfaceGladeGtk
extends java.lang.Object

FIELDS

- public static final String RADIO_GROUP_PREFIX
  –
- public static final double GTK_ALIGNMENT_LEFT_INTERVAL_MIN
  –
- public static final double GTK_ALIGNMENT_LEFT_INTERVAL_MAX
  –
- public static final double GTK_ALIGNMENT_RIGHT_INTERVAL_MIN
  –
- public static final double GTK_ALIGNMENT_RIGHT_INTERVAL_MAX
  –
- public static final double GTK_ALIGNMENT_TOP_INTERVAL_MIN
  –
- public static final double GTK_ALIGNMENT_TOP_INTERVAL_MAX
  –
- public static final double GTK_ALIGNMENT_BOTTOM_INTERVAL_MIN
  –
- public static final double GTK_ALIGNMENT_BOTTOM_INTERVAL_MAX
  –

  –

- public static final double GTK_ANGLE_HORIZONTAL_INTERVAL_MAX
  –
- public static final double GTK_ANGLE_VERTICAL_INTERVAL_MIN
  –
- public static final double GTK_ANGLE_VERTICAL_INTERVAL_MAX
  –
- public static final double GTK_ANGLE_HORIZONTAL_REVERSED_INTERVAL_MIN
  –
- public static final double GTK_ANGLE_HORIZONTAL_REVERSED_INTERVAL_MAX
  –
- public static final double GTK_ANGLE_VERTICAL_REVERSED_INTERVAL_MIN
  –
- public static final double GTK_ANGLE_VERTICAL_REVERSED_INTERVAL_MAX
  –
- public static final String GTK_JUSTIFICATION_LEFT
  –
- public static final String GTK_JUSTIFICATION_RIGHT
  –
- public static final String GTK_JUSTIFICATION_CENTER
  –
- public static final String GTK_JUSTIFICATION_FILL
  –
- public static final String GTK_POLICY_ALWAYS
  –
- public static final String GTK_POLICY_AUTOMATIC
  –
- public static final String GTK_POLICY_NEVER
  –
- public static final String GTK_SHADOW_NONE
  –
- public static final String GTK_SHADOW_IN
  –

  –

- public static final String GTK_SHADOW_ETCHED_IN
  –
- public static final String GTK_SHADOW_ETCHED_OUT
  –
- public static final String GTK_CORNER_TOP_LEFT
  –
- public static final String GTK_CORNER_BOTTOM_LEFT
  –
- public static final String GTK_CORNER_TOP_RIGHT
  –
- public static final String GTK_CORNER_BOTTOM_RIGHT
  –
- public static final String GTK_TYPE_LABEL_ITEM
  –
- public static final String GTK_TYPE_LABEL_TAB
  –
- public static final int GTK_BUTTON_TYPE_PUSH
  –
- public static final int GTK_BUTTON_TYPE_TOGGLE
  –
- public static final int GTK_BUTTON_TYPE_CHECK
  –
- public static final int GTK_BUTTON_TYPE_RADIO
  –
- public static final String GTK_GRIDOPTIONS_FILL
  –
- public static final String GTK_GRIDOPTIONS_EXPAND
  –
- public static final String GTK_GRIDOPTIONS_SHRINK
  –
- public static final String GTK_ORIENTATION_HORIZONTAL
  –

  –

- public static final String GTK_PROGRESS_LEFT_TO_RIGHT
  –
- public static final String GTK_PROGRESS_RIGHT_TO_LEFT
  –
- public static final String GTK_PROGRESS_TOP_TO_BOTTOM
  –
- public static final String GTK_PROGRESS_BOTTOM_TO_TOP
  –
- public static final String GTK_FILE_CHOOSER_ACTION_OPEN
  –
- public static final String GTK_FILE_CHOOSER_ACTION_SAVE
  –
- public static final String GTK_PACK_END
  –
- public static final String GTK_BOOLEAN_VALUE_YES
  –
- public static final String GTK_BOOLEAN_VALUE_NO
  –
- public static final String GTK_BUTTONBOX_LAYOUT_STYLE_DEFAULT
  –
- public static final String GTK_BUTTONBOX_LAYOUT_STYLE_EDGE
  –
- public static final String GTK_BUTTONBOX_LAYOUT_STYLE_START
  –
- public static final String GTK_BUTTONBOX_LAYOUT_STYLE_END
  –
- public static final String GTK_BUTTONBOX_LAYOUT_STYLE_SPREAD
  –
- public static final String GTK_PACKING_PADDING_PROPERTYNAME
  – Name of the padding GtkPacking property
- public static final String GTK_PACKING_EXPAND_PROPERTYNAME
  – Name of the expand GtkPacking property

  - Name of the fill GtkPacking property
- public static final String GTK_PACKING_TYPE_PROPERTYNAME
  - Name of the type GtkPacking property
- public static final String GTK_PACKING_TOPATTACH_PROPERTYNAME
  - Name of the top-attach GtkPacking property
- public static final String GTK_PACKING_BOTTOMATTACH_PROPERTYNAME
  - Name of the bottom-attach GtkPacking property
- public static final String GTK_PACKING_LEFTATTACH_PROPERTYNAME
  - Name of the left-attach GtkPacking property
- public static final String GTK_PACKING_RIGHTATTACH_PROPERTYNAME
  - Name of the right-attach GtkPacking property
- public static final String GTK_PACKING_TABEXPAND_PROPERTYNAME
  - Name of the tab-expand GtkPacking property
- public static final String GTK_PACKING_XOPTIONS_PROPERTYNAME
  - Name of the x-options GtkPacking property
- public static final String GTK_PACKING_YOPTIONS_PROPERTYNAME
  - Name of the y-options GtkPacking property
- public static final String GTK_PACKING_HOMOGENEOUS_PROPERTYNAME
  - Name of the homogeneous GtkPacking property
- public static final String GTK_PACKING_X_PROPERTYNAME
  - Name of the x GtkPacking property
- public static final String GTK_PACKING_Y_PROPERTYNAME
  - Name of the y GtkPacking property
- public static final String GTK_PACKING_PACKTYPE_PROPERTYNAME
  - Name of the pack-type GtkPacking property
- public static final String GTK_WIDGET_HEIGHT_REQUEST_PROPERTYNAME
  - Name of the height-request GtkWidget property
- public static final String GTK_WIDGET_WIDTH_REQUEST_PROPERTYNAME
  - Name of the width-request GtkWidget property
- public static final String GTK_WIDGET_BORDER_WIDTH_PROPERTYNAME
  - Name of the border-width GtkWidget property
- public static final String GTK_WIDGET_YALIGN_PROPERTYNAME
  - Name of the yalign GtkWidget property

  - Name of the title GtkWindow property
- public static final String GTK_WINDOW_VISIBLE_PROPERTYNAME
  - Name of the visible GtkWindow property
- public static final String GTK_WINDOW_ICON_PROPERTYNAME
  - Name of the icon GtkWindow property
- public static final String GTK_WINDOW_DEFAULT_WIDTH_PROPERTYNAME
  - Name of the default-width GtkWindow property
- public static final String GTK_WINDOW_DEFAULT_HEIGHT_PROPERTYNAME
  - Name of the default-height GtkWindow property
- public static final String GTK_SIMPLEBOX_BORDER_WIDTH_PROPERTYNAME
  - Name of the border-width GtkBox property
- public static final String GTK_SIMPLEBOX_VISIBLE_PROPERTYNAME
  - Name of the visible GtkBox property
- public static final String GTK_SIMPLEBOX_HOMOGENEOUS_PROPERTYNAME
  - Name of the homogeneous GtkBox property
- public static final String GTK_SIMPLEBOX_SPACING_PROPERTYNAME
  - Name of the spacing GtkBox property
- public static final String GTK_HBUTTONBOX_LAYOUT_STYLE_PROPERTYNAME
  - Name of the layout-style GtkHButtonBox property
- public static final String GTK_TABLEBOX_BORDER_WIDTH_PROPERTYNAME
  - Name of the border-width GtkTable property
- public static final String GTK_TABLEBOX_VISIBLE_PROPERTYNAME
  - Name of the visible GtkTable property
- public static final String GTK_TABLEBOX_HOMOGENEOUS_PROPERTYNAME
  - Name of the homogeneous GtkTable property
- public static final String GTK_TABLEBOX_COLUMNSPACING_PROPERTYNAME
  - Name of the column-spacing GtkTable property
- public static final String GTK_TABLEBOX_ROWSPACING_PROPERTYNAME
  - Name of the row-spacing GtkTable property
- public static final String GTK_TABLEBOX_COLUMNNUMBER_PROPERTYNAME
  - Name of the column-number GtkTable property
- public static final String GTK_TABLEBOX_ROWNUMBER_PROPERTYNAME
  - Name of the row-number GtkTable property

  - Name of the border-width GtkFrame property

- public static final String GTK_FRAME_VISIBLE_PROPERTYNAME
  - Name of the visible GtkFrame property
- public static final String GTK_FRAME_LABELXALIGN_PROPERTYNAME
  - Name of the label-xalign GtkBox property
- public static final String GTK_FRAME_LABELYALIGN_PROPERTYNAME
  - Name of the label-yalign GtkBox property
- public static final String GTK_FRAME_SHADOWTYPE_PROPERTYNAME
  - Name of the shadow-type GtkBox property
- public static final String GTK_SCROLLEDBOX_VISIBLE_PROPERTYNAME
  - Name of the visible GtkScrolledBox property
- public static final String GTK_SCROLLEDBOX_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkScrolledBox property
- public static final String GTK_SCROLLEDBOX_HSCROLLBAR_POLICY_PROPERTYNAME
  - Name of the hscrollbar-policy GtkScrolledBox property
- public static final String GTK_SCROLLEDBOX_VSCROLLBAR_POLICY_PROPERTYNAME
  - Name of the vscrollbar-policy GtkScrolledBox property
- public static final String GTK_SCROLLEDBOX_SHADOWTYPE_PROPERTYNAME
  - Name of the shadow-type GtkScrolledBox property
- public static final String GTK_SCROLLEDBOX_WINDOWPLACEMENT_PROPERTYNAME
  - Name of the window-placement GtkScrolledBox property
- public static final String GTK_IMAGE_PIXBUF_PROPERTYNAME
  - Name of the pixbuf (path to the image file) GtkImage property
- public static final String GTK_IMAGE_VISIBLE_PROPERTYNAME
  - Name of the visible (path to the image file) GtkImage property
- public static final String GTK_IMAGE_XPAD_PROPERTYNAME
  - Name of the xpad (path to the image file) GtkImage property
- public static final String GTK_IMAGE_YPAD_PROPERTYNAME
  - Name of the ypad (path to the image file) GtkImage property
- public static final String GTK_IMAGE_XALIGN_PROPERTYNAME
  - Name of the xalign (path to the image file) GtkImage property

  - Name of the yalign (path to the image file) GtkImage property
- public static final String GTK_LABEL_VISIBLE_PROPERTYNAME
  - Name of the visible GtkLabel property
- public static final String GTK_LABEL_LABEL_PROPERTYNAME
  - Name of the label GtkLabel property
- public static final String GTK_LABEL_USEUNDERLINE_PROPERTYNAME
  - Name of the use-underline GtkLabel property
- public static final String GTK_LABEL_USEMARKUP_PROPERTYNAME
  - Name of the use-markup GtkLabel property
- public static final String GTK_LABEL_JUSTIFY_PROPERTYNAME
  - Name of the justify GtkLabel property
- public static final String GTK_LABEL_WRAP_PROPERTYNAME
  - Name of the wrap GtkLabel property
- public static final String GTK_LABEL_SHADOWTYPE_PROPERTYNAME
  - Name of the shadow-type GtkLabel property
- public static final String GTK_LABEL_SELECTABLE_PROPERTYNAME
  - Name of the selectable GtkLabel property
- public static final String GTK_LABEL_XALIGN_PROPERTYNAME
  - Name of the xalign GtkLabel property
- public static final String GTK_LABEL_YALIGN_PROPERTYNAME
  - Name of the yalign GtkLabel property
- public static final String GTK_LABEL_XPAD_PROPERTYNAME
  - Name of the xpad GtkLabel property
- public static final String GTK_LABEL_YPAD_PROPERTYNAME
  - Name of the ypad GtkLabel property
- public static final String GTK_LABEL_ELLIPSIZE_PROPERTYNAME
  - Name of the ellipsize GtkLabel property
- public static final String GTK_LABEL_WIDTHCHARS_PROPERTYNAME
  - Name of the width-chars GtkLabel property
- public static final String GTK_LABEL_SINGLELINEMODE_PROPERTYNAME
  - Name of the single-line-mode GtkLabel property
- public static final String GTK_LABEL_ANGLE_PROPERTYNAME
  - Name of the angle GtkLabel property

  - Name of the visible GtkEntry property
- public static final String GTK_ENTRY_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkEntry property
- public static final String GTK_ENTRY_EDITABLE_PROPERTYNAME
  - Name of the editable GtkEntry property
- public static final String GTK_ENTRY_VISIBILITY_PROPERTYNAME
  - Name of the visibility GtkEntry property
- public static final String GTK_ENTRY_MAXLEGTH_PROPERTYNAME
  - Name of the max-length GtkEntry property
- public static final String GTK_ENTRY_WIDTHCHARS_PROPERTYNAME
  - Name of the width-chars GtkEntry property
- public static final String GTK_ENTRY_TEXT_PROPERTYNAME
  - Name of the text GtkEntry property
- public static final String GTK_ENTRY_HASFRAME_PROPERTYNAME
  - Name of the has-frame GtkEntry property
- public static final String GTK_ENTRY_INVISIBLECHAR_PROPERTYNAME
  - Name of the invisible-char GtkEntry property
- public static final String GTK_ENTRY_ACTIVATESDEFAULT_PROPERTYNAME
  - Name of the activates-default GtkEntry property
- public static final String GTK_TEXTVIEW_VISIBLE_PROPERTYNAME
  - Name of the visible GtkTextWiew property
- public static final String GTK_TEXTVIEW_BORDERWIDTH_PROPERTYNAME
  - Name of the visible GtkTextWiew property
- public static final String GTK_TEXTVIEW_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkEntry property
- public static final String GTK_TEXTVIEW_EDITABLE_PROPERTYNAME
  - Name of the editable GtkEntry property
- public static final String GTK_TEXTVIEW_OVERWRITE_PROPERTYNAME
  - Name of the overwrite GtkEntry property
- public static final String GTK_TEXTVIEW_ACCEPTSTAB_PROPERTYNAME
  - Name of the accepts-tab GtkEntry property
- public static final String GTK_TEXTVIEW_JUSTIFICATION_PROPERTYNAME
  - Name of the justification GtkEntry property

  - Name of the wrap-mode GtkEntry property
- public static final String GTK_TEXTVIEW_CURSORVISIBLE_PROPERTYNAME
  - Name of the cursor-visible GtkEntry property
- public static final String GTK_TEXTVIEW_PIXELSABOVE_PROPERTYNAME
  - Name of the pixels-above-lines GtkEntry property
- public static final String GTK_TEXTVIEW_PIXELSBELOW_PROPERTYNAME
  - Name of the pixels-below-lines GtkEntry property
- public static final String GTK_TEXTVIEW_PIXELSINSIDE_PROPERTYNAME
  - Name of the pixels-inside_wrap GtkEntry property
- public static final String GTK_TEXTVIEW_LEFTMARGIN_PROPERTYNAME
  - Name of the left-margin GtkEntry property
- public static final String GTK_TEXTVIEW_RIGHTMARGIN_PROPERTYNAME
  - Name of the right-margin GtkEntry property
- public static final String GTK_TEXTVIEW_INDENT_PROPERTYNAME
  - Name of the indent GtkEntry property
- public static final String GTK_TEXTVIEW_TEXT_PROPERTYNAME
  - Name of the text GtkEntry property
- public static final String GTK_BUTTON_FOCUS_ON_CLICK_PROPERTYNAME
  - Name of the focus-on-click GtkButton property
- public static final String GTK_BUTTON_IMAGE_PROPERTYNAME
  - Name of the image GtkButton property
- public static final String GTK_BUTTON_IMAGE_POSITION_PROPERTYNAME
  - Name of the image-position GtkButton property
- public static final String GTK_BUTTON_LABEL_PROPERTYNAME
  - Name of the label GtkButton property
- public static final String GTK_BUTTON_RELIEF_PROPERTYNAME
  - Name of the relief GtkButton property
- public static final String GTK_BUTTON_USE_STOCK_PROPERTYNAME
  - Name of the use-stock GtkButton property
- public static final String GTK_BUTTON_USE_UNDERLINE_PROPERTYNAME
  - Name of the use-underline GtkButton property
- public static final String GTK_BUTTON_XALIGN_PROPERTYNAME
  - Name of the xalign GtkButton property

  - Name of the yalign GtkButton property

- public static final String GTK_BUTTON_VISIBLE_PROPERTYNAME
  - Name of the visible GtkButton property
- public static final String GTK_TOGGLEBUTTON_ACTIVE_PROPERTYNAME
  - Name of the active GtkToggleButton property
- public static final String GTK_TOGGLEBUTTON_INCONSISTENT_PROPERTYNAME
  - Name of the inconsistent GtkToggleButton property
- public static final String GTK_TOGGLEBUTTON_DRAWINDICATOR_PROPERTYNAME
  - Name of the draw-indicator GtkToggleButton property
- public static final String GTK_CHECKBUTTON_INDICATORSIZE_PROPERTYNAME
  - Name of the indicator-size GtkCheckButton property
- public static final String GTK_CHECKBUTTON_INDICATORSPACING_PROPERTYNAME
  - Name of the indicator-spacing GtkCheckButton property
- public static final String GTK_RADIOBUTTON_GROUP_PROPERTYNAME
  - Name of the group GtkRadioButton property
- public static final String GTK_COMBOBOX_VISIBLE_PROPERTYNAME
  - Name of the visible GtkComboBox property
- public static final String GTK_COMBOBOX_ITEMS_PROPERTYNAME
  - Name of the items GtkComboBox property
- public static final String GTK_COMBOBOX_ADDTEAROFFS_PROPERTYNAME
  - Name of the add-tearoffs GtkComboBox property
- public static final String GTK_COMBOBOX_FOCUS_ON_CLICK_PROPERTYNAME
  - Name of the focus-on-click GtkComboBox property
- public static final String GTK_COMBOBOX_ACTIVE_PROPERTYNAME
  - Name of the active GtkComboBox property
- public static final String GTK_COMBOBOX_HASFRAME_PROPERTYNAME
  - Name of the has-frame GtkComboBox property
- public static final String GTK_COMBOBOX_COLUMNSPAN_PROPERTYNAME
  - Name of the column-span-column GtkComboBox property
- public static final String GTK_COMBOBOX_ROWSPAN_PROPERTYNAME
  - Name of the row-span-column GtkComboBox property

  - Name of the model GtkComboBox property
- public static final String GTK_COMBOBOX_WRAPWIDTH_PROPERTYNAME
  - Name of the wrap-width GtkComboBox property
- public static final String GTK_COMBOBOX_APPEARS_AS_LIST_PROPERTYNAME
  - Name of the appears-as-list GtkComboBox property
- public static final String GTK_SPINBUTTON_VISIBLE_PROPERTYNAME
  - Name of the visible GtkSpinButton property
- public static final String GTK_SPINBUTTON_ADJUSTMENT_PROPERTYNAME
  - Name of the adjustment GtkSpinButton property
- public static final String GTK_SPINBUTTON_CLIMBRATE_PROPERTYNAME
  - Name of the climb-rate GtkSpinButton property
- public static final String GTK_SPINBUTTON_DIGITS_PROPERTYNAME
  - Name of the digits GtkSpinButton property
- public static final String GTK_SPINBUTTON_NUMERIC_PROPERTYNAME
  - Name of the numeric GtkSpinButton property
- public static final String GTK_SPINBUTTON_SNAPTOTICKS_PROPERTYNAME
  - Name of the snap-to-ticks GtkSpinButton property
- public static final String GTK_SPINBUTTON_UPDATEPOLICY_PROPERTYNAME
  - Name of the update-policy GtkSpinButton property
- public static final String GTK_SPINBUTTON_VALUE_PROPERTYNAME
  - Name of the value GtkSpinButton property
- public static final String GTK_SPINBUTTON_WRAP_PROPERTYNAME
  - Name of the wrap GtkSpinButton property
- public static final String GTK_SPINBUTTON_SHADOWTYPE_PROPERTYNAME
  - Name of the shadow-type GtkSpinButton property
- public static final String GTK_STATUSBAR_VISIBLE_PROPERTYNAME
  - Name of the visible GtkStatusBar property
- public static final String GTK_STATUSBAR_BORDERWIDTH_PROPERTYNAME
  - Name of the border-width GtkStatusBar property
- public static final String GTK_STATUSBAR_HASRESIZEGRIP_PROPERTYNAME
  - Name of the has-resize-grip GtkStatusBar property
- public static final String GTK_NOTEBOOK_VISIBLE_PROPERTYNAME
  - Name of the visible GtkNoteBook property

  - Name of the can-focus GtkNoteBook property
- public static final String GTK_NOTEBOOK_SHOWTABS_PROPERTYNAME
  - Name of the show-tabs GtkNoteBook property
- public static final String GTK_NOTEBOOK_SHOWBORDER_PROPERTYNAME
  - Name of the show-border GtkNoteBook property
- public static final String GTK_NOTEBOOK_TABPOS_PROPERTYNAME
  - Name of the tab-pos GtkNoteBook property
- public static final String GTK_NOTEBOOK_SCROLLABLE_PROPERTYNAME
  - Name of the scrollable GtkNoteBook property
- public static final String GTK_NOTEBOOK_ENABLEPOPUP_PROPERTYNAME
  - Name of the enable-popup GtkNoteBook property
- public static final String GTK_MENUBAR_VISIBLE_PROPERTYNAME
  - Name of the visible GtkMenuBar property
- public static final String GTK_MENUBAR_PACKDIRECTION_PROPERTYNAME
  - Name of the pack-direction GtkMenuBar property
- public static final String GTK_MENUBAR_CHILD_PACK_DIRECTION_PROPERTYNAME
  - Name of the child-pack-direction GtkMenuBar property
- public static final String GTK_MENUBAR_BORDERWIDTH_PROPERTYNAME
  - Name of the border-width GtkMenuBar property
- public static final String GTK_MENUITEM_VISIBLE_PROPERTYNAME
  - Name of the visible GtkMenuItem property
- public static final String GTK_MENUITEM_USESTOCK_PROPERTYNAME
  - Name of the use-stock GtkMenuItem property
- public static final String GTK_MENUITEM_USEUNDERLINE_PROPERTYNAME
  - Name of the use-underline GtkMenuItem property
- public static final String GTK_MENUITEM_LABEL_PROPERTYNAME
  - Name of the label GtkMenuItem property
- public static final String GTK_MENUITEM_ARROWSPACING_PROPERTYNAME
  - Name of the arrow-spacing GtkMenuItem property
- public static final String GTK_MENUITEM_HORIZONTALPADDING_PROPERTYNAME
  - Name of the horizontal-padding GtkMenuItem property
- public static final String

  - Name of the selected-shadow-type GtkMenuItem property

- public static final String GTK_MENUITEM_TOGGLESPACING_PROPERTYNAME
  - Name of the toggle-spacing GtkMenuItem property
- public static final String GTK_CALENDAR_VISIBLE_PROPERTYNAME
  - Name of the visible GtkCalendar property
- public static final String GTK_CALENDAR_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkCalendar property
- public static final String GTK_CALENDAR_DISPLAYOPTIONS_PROPERTYNAME
  - Name of the display-options GtkCalendar property
- public static final String GTK_TOOLBAR_VISIBLE_PROPERTYNAME
  - Name of the visible GtkToolBar property
- public static final String GTK_TOOLBAR_ORIENTATION_PROPERTYNAME
  - Name of the orientation GtkToolBar property
- public static final String GTK_TOOLBAR_STYLE_PROPERTYNAME
  - Name of the toolbar_style GtkToolBar property
- public static final String GTK_TOOLBAR_TOOLTIPS_PROPERTYNAME
  - Name of the tooltips GtkToolBar property
- public static final String GTK_TOOLBAR_SHOWARROW_PROPERTYNAME
  - Name of the show-arrow GtkToolBar property
- public static final String GTK_TOOLBUTTON_VISIBLE_PROPERTYNAME
  - Name of the visible GtkToolButton property
- public static final String GTK_TOOLBUTTON_VISIBLE_HORIZONTAL_PROPERTYNAME
  - Name of the visible-horizontal GtkToolButton property
- public static final String GTK_TOOLBUTTON_VISIBLE_VERTICAL_PROPERTYNAME
  - Name of the visible-vertical GtkToolButton property
- public static final String GTK_TOOLBUTTON_LABEL_PROPERTYNAME
  - Name of the label GtkToolButton property
- public static final String GTK_TOOLBUTTON_USEUNDERLINE_PROPERTYNAME
  - Name of the use-underline GtkToolButton property
- public static final String GTK_TOOLBUTTON_IS_IMPORTANT_PROPERTYNAME
  - Name of the is-important GtkToolButton property
- public static final String GTK_TOOLBUTTON_ICON_PROPERTYNAME

- public static final String GTK_TOGGLETOOLBUTTON_ACTIVE_PROPERTYNAME
    - Name of the active GtkToggleToolButton property
- public static final String GTK_SEPARATORTOOLITEM_DRAW_PROPERTYNAME
    - Name of the draw GtkToolButton property
- public static final String GTK_RADIOTOOLBUTTON_GROUP_PROPERTYNAME
    - Name of the group GtkRadioToolButton property
- public static final String GTK_TOOLITEM_VISIBLE_PROPERTYNAME
    - Name of the visible GtkToolItem property
- public static final String GTK_TOOLITEM_VISIBLE_HORIZONTAL_PROPERTYNAME
    - Name of the visible-horizontal GtkToolItem property
- public static final String GTK_TOOLITEM_VISIBLE_VERTICAL_PROPERTYNAME
    - Name of the visible-vertical GtkToolItem property
- public static final String GTK_TOOLITEM_IS_IMPORTANT_PROPERTYNAME
    - Name of the is-important GtkToolItem property
- public static final String GTK_PANE_VISIBLE_PROPERTYNAME
    - Name of the visible GtkPane property
- public static final String GTK_PANE_CANFOCUS_PROPERTYNAME
    - Name of the can-focus GtkPane property
- public static final String GTK_PANE_POSITION_PROPERTYNAME
    - Name of the position GtkPane property
- public static final String GTK_DRAWINGAREA_VISIBLE_PROPERTYNAME
    - Name of the visible GtkDrawingArea property
- public static final String GTK_FILECHOOSER_DIALOG_VISIBLE_PROPERTYNAME
    - Name of the visible GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_TITLE_PROPERTYNAME
    - Name of the title GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_ACTION_PROPERTYNAME
    - Name of the action GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_LOCALONLY_PROPERTYNAME
    - Name of the local-only GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_SELECTMULTIPLE_PROPERTYNAME
    - Name of the select-multiple GtkFileChooserDialog property

- public static final String GTK_FILECHOOSER_DIALOG_SHOWHIDDEN_PROPERTYNAME
  - Name of the show-hidden GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_DOOVERWRITECONFIRMATION_PROPERTYNAME
  - Name of the do-overwrite-confirmation GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_TYPE_PROPERTYNAME
  - Name of the type GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_WINDOWPOSITION_PROPERTYNAME
  - Name of the window-position GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_MODAL_PROPERTYNAME
  - Name of the modal GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_RESIZABLE_PROPERTYNAME
  - Name of the resizable GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_DESTROYWITHPARENT_PROPERTYNAME
  - Name of the destroy-with-parent GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_DECORATED_PROPERTYNAME
  - Name of the decorated GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_SKIPTASKBARHINT_PROPERTYNAME
  - Name of the skip-taskbar GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_SKIPPAGERHINT_PROPERTYNAME
  - Name of the skip-pager-hint GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_TYPEHINT_PROPERTYNAME
  - Name of the type-hint GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_GRAVITY_PROPERTYNAME
  - Name of the gravity GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_FOCUSONMAP_PROPERTYNAME
  - Name of the focus-on-map GtkFileChooserDialog property
- public static final String GTK_FILECHOOSER_DIALOG_URGENCYHINT_PROPERTYNAME

- public static final String GTK_COLORSELECTION_DIALOG_VISIBLE_PROPERTYNAME
  - Name of the visible GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_TITLE_PROPERTYNAME
  - Name of the title GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_TYPE_PROPERTYNAME
  - Name of the type GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_WINDOWPOSITION_PROPERTYNAME
  - Name of the window-position GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_MODAL_PROPERTYNAME
  - Name of the modal GtkColorSelectionrDialog property
- public static final String GTK_COLORSELECTION_DIALOG_RESIZABLE_PROPERTYNAME
  - Name of the resizable GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_DESTROYWITHPARENT_PROPERTYNAME
  - Name of the destroy-with-parent GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_DECORATED_PROPERTYNAME
  - Name of the decorated GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_SKIPTASKBARHINT_PROPERTYNAME
  - Name of the skip-taskbar GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_SKIPPAGERHINT_PROPERTYNAME
  - Name of the skip-pager-hint GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_TYPEHINT_PROPERTYNAME
  - Name of the type-hint GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_GRAVITY_PROPERTYNAME
  - Name of the gravity GtkColorSelectionDialog property
- public static final String GTK_COLORSELECTION_DIALOG_FOCUSONMAP_PROPERTYNAME
  - Name of the [illegible] GtkColorSelectionDialog property

- public static final String GTK_COLORSELECTION_DIALOG_URGENCYHINT_PROPERTYNAME
  - Name of the urgency-hint GtkColorSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_VISIBLE_PROPERTYNAME
  - Name of the visible GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_TITLE_PROPERTYNAME
  - Name of the title GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_TYPE_PROPERTYNAME
  - Name of the type GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_WINDOWPOSITION_PROPERTYNAME
  - Name of the window-position GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_MODAL_PROPERTYNAME
  - Name of the modal GtkFontSelectionrDialog property
- public static final String GTK_FONTSELECTION_DIALOG_RESIZABLE_PROPERTYNAME
  - Name of the resizable GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_DESTROYWITHPARENT_PROPERTYNAME
  - Name of the destroy-with-parent GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_DECORATED_PROPERTYNAME
  - Name of the decorated GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_SKIPTASKBARHINT_PROPERTYNAME
  - Name of the skip-taskbar GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_SKIPPAGERHINT_PROPERTYNAME
  - Name of the skip-pager-hint GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_TYPEHINT_PROPERTYNAME
  - Name of the type-hint GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_GRAVITY_PROPERTYNAME
  - Name of the gravity GtkFontSelectionDialog property
- public static final String

  - Name of the focus-on-map GtkFontSelectionDialog property
- public static final String GTK_FONTSELECTION_DIALOG_URGENCYHINT_PROPERTYNAME
  - Name of the urgency-hint GtkFontSelectionDialog property
- public static final String GTK_DIALOG_VISIBLE_PROPERTYNAME
  - Name of the visible GtkDialog property
- public static final String GTK_DIALOG_TITLE_PROPERTYNAME
  - Name of the title GtkDialog property
- public static final String GTK_DIALOG_TYPE_PROPERTYNAME
  - Name of the type GtkDialog property
- public static final String GTK_DIALOG_WINDOWPOSITION_PROPERTYNAME
  - Name of the window-position GtkDialog property
- public static final String GTK_DIALOG_MODAL_PROPERTYNAME
  - Name of the modal GtkDialog property
- public static final String GTK_DIALOG_RESIZABLE_PROPERTYNAME
  - Name of the resizable GtkDialog property
- public static final String GTK_DIALOG_DESTROYWITHPARENT_PROPERTYNAME
  - Name of the destroy-with-parent GtkDialog property
- public static final String GTK_DIALOG_DECORATED_PROPERTYNAME
  - Name of the decorated GtkDialog property
- public static final String GTK_DIALOG_SKIPTASKBARHINT_PROPERTYNAME
  - Name of the skip-taskbar GtkDialog property
- public static final String GTK_DIALOG_SKIPPAGERHINT_PROPERTYNAME
  - Name of the skip-pager-hint GtkDialog property
- public static final String GTK_DIALOG_TYPEHINT_PROPERTYNAME
  - Name of the type-hint GtkDialog property
- public static final String GTK_DIALOG_GRAVITY_PROPERTYNAME
  - Name of the gravity GtkDialog property
- public static final String GTK_DIALOG_FOCUSONMAP_PROPERTYNAME
  - Name of the focus-on-map GtkDialog property
- public static final String GTK_DIALOG_URGENCYHINT_PROPERTYNAME
  - Name of the urgency-hint GtkDialog property
- public static final String GTK_FIXED_VISIBLE_PROPERTYNAME

- public static final String GTK_PROGRESSBAR_VISIBLE_PROPERTYNAME
  - Name of the visible GtkProgressBar property
- public static final String GTK_PROGRESSBAR_ORIENTATION_PROPERTYNAME
  - Name of the orientation GtkProgressBar property
- public static final String GTK_PROGRESSBAR_FRACTION_PROPERTYNAME
  - Name of the fraction GtkProgressBar property
- public static final String GTK_PROGRESSBAR_PULSESTEP_PROPERTYNAME
  - Name of the pulse-step GtkProgressBar property
- public static final String GTK_PROGRESSBAR_ELLIPSIZE_PROPERTYNAME
  - Name of the ellipsize GtkProgressBar property
- public static final String GTK_SCALE_VISIBLE_PROPERTYNAME
  - Name of the visible GtkScale property
- public static final String GTK_SCALE_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkScale property
- public static final String GTK_SCALE_DRAWVALUE_PROPERTYNAME
  - Name of the draw-value GtkScale property
- public static final String GTK_SCALE_VALUEPOS_PROPERTYNAME
  - Name of the value-pos GtkScale property
- public static final String GTK_SCALE_DIGITS_PROPERTYNAME
  - Name of the digits GtkScale property
- public static final String GTK_SCALE_UPDATEPOLICY_PROPERTYNAME
  - Name of the update-policy GtkScale property
- public static final String GTK_SCALE_INVERTED_PROPERTYNAME
  - Name of the inverted GtkScale property
- public static final String GTK_SCALE_ADJUSTMENT_PROPERTYNAME
  - Name of the adjustment GtkScale property
- public static final String GTK_TREEVIEW_VISIBLE_PROPERTYNAME
  - Name of the visible GtkTreeView property
- public static final String GTK_TREEVIEW_CANFOCUS_PROPERTYNAME
  - Name of the can-focus GtkTreeView property
- public static final String GTK_TREEVIEW_HEADERSVISIBLE_PROPERTYNAME
  - Name of the headers-visible GtkTreeView property
- public static final String GTK_TREEVIEW_RULESHINT_PROPERTYNAME

- public static final String GTK_TREEVIEW_REORDERABLE_PROPERTYNAME
  - Name of the reorderable GtkTreeView property
- public static final String GTK_TREEVIEW_ENABLESEARCH_PROPERTYNAME
  - Name of the enable-search GtkTreeView property
- public static final String GTK_TREEVIEW_FIXEDHEIGHT_MODE_PROPERTYNAME
  - Name of the fixed-height-mode GtkTreeView property
- public static final String GTK_TREEVIEW_HOVERSELECTION_PROPERTYNAME
  - Name of the hover-selection GtkTreeView property
- public static final String GTK_TREEVIEW_HOVEREXPAND_PROPERTYNAME
  - Name of the hover-expand GtkTreeView property
- public static final String GTK_TREEITEM_TEXT_PROPERTYNAME
  - Name of the text GtkTreeItem property
- public static final String GTK_TREEITEM_ICON_PROPERTYNAME
  - Name of the icon GtkTreeItem property
- public static final String GTK_WIDGET_TYPE_WINDOW
  -
- public static final String GTK_WIDGET_TYPE_DIALOG
  -
- public static final String GTK_WIDGET_TYPE_FILECHOOSERDIALOG
  -
- public static final String GTK_WIDGET_TYPE_COLORSELECTIONDIALOG
  -
- public static final String GTK_WIDGET_TYPE_FONTSELECTIONDIALOG
  -
- public static final String GTK_WIDGET_TYPE_VBOX
  -
- public static final String GTK_WIDGET_TYPE_HBOX
  -
- public static final String GTK_WIDGET_TYPE_VBUTTONBOX
  -
- public static final String GTK_WIDGET_TYPE_HBUTTONBOX
  -
- public static final String GTK_WIDGET_TYPE_BUTTON

- public static final String GTK_WIDGET_TYPE_CHECKBUTTON
  –
- public static final String GTK_WIDGET_TYPE_PUSHBUTTON
  –
- public static final String GTK_WIDGET_TYPE_RADIOBUTTON
  –
- public static final String GTK_WIDGET_TYPE_TOGGLEBUTTON
  –
- public static final String GTK_WIDGET_TYPE_ENTRY
  –
- public static final String GTK_WIDGET_TYPE_FRAME
  –
- public static final String GTK_WIDGET_TYPE_TABLE
  –
- public static final String GTK_WIDGET_TYPE_LABEL
  –
- public static final String GTK_WIDGET_TYPE_IMAGE
  –
- public static final String GTK_WIDGET_TYPE_TEXTVIEW
  –
- public static final String GTK_WIDGET_TYPE_ALIGNMENT
  –
- public static final String GTK_WIDGET_TYPE_SCROLLEDWINDOW
  –
- public static final String GTK_WIDGET_TYPE_COMBOBOX
  –
- public static final String GTK_WIDGET_TYPE_COMBOBOXENTRY
  –
- public static final String GTK_WIDGET_TYPE_SPINBUTTON
  –
- public static final String GTK_WIDGET_TYPE_STATUSBAR
  –
- public static final String GTK_WIDGET_TYPE_NOTEBOOK

- public static final String GTK_WIDGET_TYPE_MENUBAR
  –
- public static final String GTK_WIDGET_TYPE_MENU
  –
- public static final String GTK_WIDGET_TYPE_MENUITEM
  –
- public static final String GTK_WIDGET_TYPE_MENUIMAGEITEM
  –
- public static final String GTK_WIDGET_TYPE_SEPARATORMENUITEM
  –
- public static final String GTK_WIDGET_TYPE_CALENDAR
  –
- public static final String GTK_WIDGET_TYPE_TOOLBAR
  –
- public static final String GTK_WIDGET_TYPE_TOOLBUTTON
  –
- public static final String GTK_WIDGET_TYPE_TOOLRADIOBUTTON
  –
- public static final String GTK_WIDGET_TYPE_TOOLTOGGLEBUTTON
  –
- public static final String GTK_WIDGET_TYPE_TOOLSEPARATOR
  –
- public static final String GTK_WIDGET_TYPE_TOOLITEM
  –
- public static final String GTK_WIDGET_TYPE_VIEWPORT
  –
- public static final String GTK_WIDGET_TYPE_VPANED
  –
- public static final String GTK_WIDGET_TYPE_HPANED
  –
- public static final String GTK_WIDGET_TYPE_VSEPARATOR
  –
- public static final String GTK_WIDGET_TYPE_HSEPARATOR

- public static final String GTK_WIDGET_TYPE_DRAWINGAREA
  -
- public static final String GTK_WIDGET_TYPE_FIXED
  -
- public static final String GTK_WIDGET_TYPE_PROGRESSBAR
  -
- public static final String GTK_WIDGET_TYPE_VSCALE
  -
- public static final String GTK_WIDGET_TYPE_HSCALE
  -
- public static final String GTK_WIDGET_TYPE_TREEVIEW
  -
- public static final String GTK_SIGNAL_TYPE_CLICKED
  -
- public static final String GTK_SIGNAL_TYPE_DOUBLECLICKED
  -
- public static final String GTK_SIGNAL_TYPE_ENTER
  -
- public static final String GTK_SIGNAL_TYPE_LEAVE
  -
- public static final String GTK_SIGNAL_TYPE_MOVECURSOR
  -
- public static final String GTK_SIGNAL_TYPE_MOVEFOCUSOUT
  -
- public static final String GTK_SIGNAL_TYPE_FOCUSIN
  -
- public static final String GTK_SIGNAL_TYPE_FOCUSOUT
  -
- public static final String GTK_SIGNAL_TYPE_HIDE
  -
- public static final String GTK_SIGNAL_TYPE_KEYPRESSED
  -
- public static final String GTK_SIGNAL_TYPE_KEYRELEASED

- public static final String GTK_SIGNAL_TYPE_SHOW
  –
- public static final String GTK_SIGNAL_TYPE_DESTROY
  –
- public static final String GTK_SIGNAL_TYPE_DELETE
  –
- public static final String GTK_SIGNAL_TYPE_SIZEREQUEST
  –
- public static final String GTK_SIGNAL_TYPE_CHECKRESIZE
  –
- public static final String GTK_SIGNAL_TYPE_INSERTATCURSOR
  –
- public static final String GTK_SIGNAL_TYPE_DELETEFROMCURSOR
  –
- public static final String GTK_SIGNAL_TYPE_CHANGED
  –
- public static final String GTK_SIGNAL_TYPE_DELETETEXT
  –
- public static final String GTK_SIGNAL_TYPE_INSERTTEXT
  –
- public static final String GTK_SIGNAL_TYPE_CHANGEVALUE
  –
- public static final String GTK_SIGNAL_TYPE_VALUECHANGED
  –
- public static final String GTK_SIGNAL_TYPE_SWITCHPAGE
  –
- public static final String GTK_SIGNAL_TYPE_CHANGECURRENTPAGE
  –
- public static final String GTK_SIGNAL_TYPE_SELECTPAGE
  –
- public static final String GTK_SIGNAL_TYPE_ACTIVATE
  –
- public static final String GTK_SIGNAL_TYPE_PRESSED

- public static final String GTK_SIGNAL_TYPE_RELEASED
  –
- public static final String GTK_SIGNAL_TYPE_TOGGLED
  –
- public static final String GTK_SIGNAL_TYPE_GROUPCHANGED
  –
- public static final String GTK_SIGNAL_TYPE_DAYSELECTED
  –
- public static final String GTK_SIGNAL_TYPE_DAYSELECTED_DOUBLECLICK
  –
- public static final String GTK_SIGNAL_TYPE_ROWCOLLAPSED
  –
- public static final String GTK_SIGNAL_TYPE_ROWEXPANDED
  –
- public static final String GTK_SIGNAL_TYPE_SELECTCURSORROW
  –
- public static final String GTK_SIGNAL_TYPE_SELECTCURSORPARENT
  –
- public static final String GTK_SIGNAL_TYPE_CLOSE
  –
- public static final String GTK_SIGNAL_TYPE_COPYCLIPBOARD
  –
- public static final String GTK_SIGNAL_TYPE_POPULATEPOPUP
  –
- public static final String GTK_SIGNAL_TYPE_SCROLLCHILD
  –
- public static final String GTK_SIGNAL_TYPE_BACKSPACE
  –
- public static final String GTK_SIGNAL_TYPE_CUTCLIPBOARD
  –
- public static final String GTK_SIGNAL_TYPE_MOVEFOCUS
  –
- public static final String GTK_SIGNAL_TYPE_MOVEVIEWPORT

- public static final String GTK_SIGNAL_TYPE_PAGEHORIZONTALLY
  –
- public static final String GTK_SIGNAL_TYPE_PASTECLIPBOARD
  –
- public static final String GTK_SIGNAL_TYPE_SELLECTALL
  –
- public static final String GTK_SIGNAL_TYPE_SETANCHOR
  –
- public static final String GTK_SIGNAL_TYPE_SETSCROLLADJUSTMENTS
  –
- public static final String GTK_SIGNAL_TYPE_TOGGLEOVERWRITE
  –
- public static final String GTK_SIGNAL_TYPE_INPUT
  –
- public static final String GTK_SIGNAL_TYPE_OUTPUT
  –

CONSTRUCTORS

- *InterfaceGladeGtk*
  `public` **`InterfaceGladeGtk`**`( )`

2.2.4 CLASS **JavaGlade**

Main class of the JavaGlade module. It gives access to the conversion methods and to the GUI widgets created as the result of the conversion. It acts as the JavaGlade interface to other applications. It's the faade to the conversion methods and gives access to the GUI widgets and objects. <br/>Responsibilities:

- Faade to JavaGlade module functionalities.</li>
- Parsing of the .glade interface files.</li>
- Conversion of the .glade interface into a Java compatible GUI.</li>
- Provide accessors for the GUI objects.</li>
- Start and stop the execution of the generated GUI.</li>
- Register GUI events and connect them to their handlers.</li>

<br/>

DECLARATION

---

public class JavaGlade
extends java.lang.Object

CONSTRUCTORS

- *JavaGlade*
  `public` **JavaGlade**`( )`
  - **Usage**
    - Default JavaGlade constructor.

- *JavaGlade*
  `public` **JavaGlade**`( java.lang.String` **guiName** `)`

- *JavaGlade*
  `public` **JavaGlade**`( java.lang.String` **guiName**, `com.imagos.javaglade.Gui` **parentGui**, `java.lang.String` **parentName** `)`

- *JavaGlade*
  `public` **JavaGlade**`( java.lang.String` **guiName**, `com.imagos.javaglade.Gui` **parentGui**, `java.lang.String` **parentName**, `Shell` **parentShell** `)`

METHODS

- *buildInterface*
  `public void` **buildInterface**`( com.imagos.javaglade.vo.GladeInterface` **iface**, `java.lang.String` **rootnode**, `Widget` **rootParent** `)`
  - **Usage**
    - Builds the interface described by iface.
  - **Parameters**
    - `iface` - GladeInterface, the interface parsed from the glade file.
    - `rootnode` - String, the root node specified (if user chose building from a particular widget and not the toplevel).
    - `rootParent` - Widget, parent widget of the parsed widgets when a root node is used different from the top-level windows.
  - **Exceptions**
    - `JavaGladeException,` - in case any error took place during the building of the GUI.

- *construct*
  `public Gui` **construct**`( com.imagos.javaglade.vo.GladeInterface` **iface**, `java.lang.String` **rootnode**, `Widget` **parent**, `java.lang.String` **domain** `)`
  - **Usage**
    - Converts the Gtk structure into a valid Java SWT widget set. Constructs all the widgets represented by the GladeInterface structure and add them to the GUI.
  - **Parameters**
    - `iface` - GladeInterface, structure with the parsed hierarchy of Gtk widgets.
    - `rootnode` - String, the root widget node (or null for none)
    - `parent` - Widget, parent widget of the parsed widgets when a root node is used different from the top-level window (or null for parsing the complete tree of

    - `domain` - String, the translation domain (or null for the default)
  - **Returns** - Gui, instance of the built GUI.
  - **Exceptions**
    - `JGladeConstructionExeption` - if construction fails

- *emptyPanel*
  `public Widget` **emptyPanel**`( Composite` **parent**`, java.lang.String` **parentName**`, java.lang.String` **name** `)`
  - **Usage**
    - Build a widget of type COMPOSITE with no components inside.
  - **Parameters**
    - `parent` - Composite, parent of the empty composite widget.
    - `name` - String, name of the widget.
  - **Returns** - Widget, the widget built.

- *getColor*
  `public Color` **getColor**`( int` **color** `)`
  - **Usage**
    - Gets a color defined by the constant.
  - **Parameters**
    - `color` - int, color constant.
  - **Returns** - Color, color requested.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

2.2.5 CLASS **JavaGladeCatalog**

Implements the catalog of widgets built by the JavaGlade conversion module. It gives access to the GUI widgets and objects. <br/>Responsibilities:

- Store the built GUI widgets.</li>
- Provide accessors for the GUI objects.</li>

<br/>

DECLARATION

public class JavaGladeCatalog
extends java.lang.Object

METHODS

- *addWidget*
  `public void` **addWidget**`( java.lang.String` **name**`,`
  `com.imagos.javaglade.toolkit.swt.SwtWidget` **widget** `)`
  - **Usage**
    - Adds a widget to the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the widget.
    - `SwtWidget` - widget, widget to add to the catalog.

- *getDialog*
  `public SwtDialog` **getDialog**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets the dialog widget from the widgets' catalog.
  - **Parameters**
    - `name` - String, widget name of the dialog to be searched.
  - **Returns** - SwtDialog, Dialog widget found (NULL in case no widget could be found).

- *getInstance*
  `public static final JavaGladeCatalog` **getInstance**`( )`
  - **Usage**
    - Returns the unique instance of the JavaGladeCatalog class. Implements the singleton dessign pattern.
  - **Returns** - JavaGladeCatalog, unique instance of JavaGladeCatalog.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getRootWidget*
  `public SwtWidget` **getRootWidget**`( java.lang.String` **guiName** `)`
  - **Usage**
    - Gets the root widget of an interface. The root widget is the toppest widget in the hierarchy (normally an SwtTopLevelWindow).
  - **Parameters**
    - `guiName,` - name of the interface to search.
  - **Returns** - SwtWidget, instance of the main widget of the interface.

- *getSwtWidget*
  `public SwtWidget` **getSwtWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - This function is used to get the Widget corresponding to the name in the interface

  - **Parameters**
    - `name` - String, the name of the widget.
  - **Returns** - the widget matching name, or null if not exist

---

- *getWidget*
  `public Widget` **getWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - This function is used to get the Widget corresponding to the name in the interface description. You would use this if you have to do anything to the widget after loading.
  - **Parameters**
    - `name` - String, the name of the widget.
  - **Returns** - the widget matching name, or null if not exist

---

- *getWindow*
  `public SwtTopLevelWindow` **getWindow**`( java.lang.String` **name** `)`
  - **Usage**
    - This function is used to get the Window corresponding to the name in the interface description. You would use this if you have to do anything to the window after loading.
  - **Parameters**
    - `name` - String, the name of the window.
  - **Returns** - SwtTopLevelWindow, the window matching the name, or null if not exist.

- *removeWidget*
  `public void` **removeWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - Removes a widget from the widgets' catalog.
  - **Parameters**
    - `name` - String, name of the widget.

- *renameWidget*
  `public void` **renameWidget**`( java.lang.String` **oldName**`, java.lang.String` **newName** `)`
  - **Usage**
    - Renames a widget in the widgets' catalog.
  - **Parameters**
    - `oldName` - String, name of the widget.
    - `newName` - String, new name given to the widget.

2.2.6 CLASS **JavaGladeException**

This class encapsulates and holds all the errors that might have taken place during any of the Glade to Java conversion operations. <br/>Responsibilities:

- Encapsulate and pass on the error messages to the upper layers. </li>

DECLARATION

public class JavaGladeException
extends java.lang.Exception

CONSTRUCTORS

- *JavaGladeException*
  `public` **JavaGladeException**`( )`
  - **Usage**
    - Default JavaGladeException constructor.

- *JavaGladeException*
  `public` **JavaGladeException**`( java.lang.String` **message** `)`
  - **Usage**
    - Constructor for the JavaGladeException with the message for the exception.
  - **Parameters**
    - `message` - String, message error.

- *JavaGladeException*
  `public` **JavaGladeException**`( java.lang.String` **message**`, java.lang.Throwable` **cause** `)`
  - **Usage**
    - Constructor for the JavaGladeException with the message for the exception and the source exception that raised the error.
  - **Parameters**
    - `message` - String, message error.
    - `cause` - Throwable, source exception.

- *JavaGladeException*
  `public` **JavaGladeException**`( java.lang.Throwable` **cause** `)`
  - **Usage**
    - Constructor for the JavaGladeException with the source exception that raised the error.
  - **Parameters**
    - `cause` - Throwable, source exception.

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*
  `public synchronized native Throwable` **fillInStackTrace**`( )`
- *getCause*
  `public Throwable` **getCause**`( )`
- *getLocalizedMessage*
  `public String` **getLocalizedMessage**`( )`
- *getMessage*
  `public String` **getMessage**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *initCause*
  `public synchronized Throwable` **initCause**`( java.lang.Throwable` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( )`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintStream` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintWriter` **arg0** `)`
- *setStackTrace*
  `public void` **setStackTrace**`( java.lang.StackTraceElement []` **arg0** `)`
- *toString*
  `public String` **toString**`( )`

2.2.7 CLASS **JavaGladeTranslator**

This class functions as a Factory for I18n instances that will be used to translate the texts into the user's current locale. <br/>

DECLARATION

public class JavaGladeTranslator
extends java.lang.Object

CONSTRUCTORS

- *JavaGladeTranslator*
  `public` **JavaGladeTranslator**`( java.util.Locale` **usrLocale**, `java.lang.Class` **cl** `)`
  - **Usage**
    - Gets the correct Internationalization instance for the required Locale and class.
  - **Parameters**
    - `usrLocale` - Locale, current user's locale data.

METHODS

- *getI18n*
  `public I18n` **`getI18n`**`( )`
  - **Usage**
    - Gets the current instance of Internationalization class.
  - **Returns** - I18n, instance of Internationalization class.

- *getRevision*
  `public static String` **`getRevision`**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

2.2.8 CLASS JavaGladeUtil

Faade of the JavaGlade module. It keeps the global constants and gives access to the main methods of the JavaGlade interface builder. <br/>Responsibilities:

- Store the global configuration values.</li>
- Interface for using the JavaGlade module as a part of a bigger application.</li>
- Gives access to the JavaGlade interface construction methods.</li>

<br/>

DECLARATION

---

public class JavaGladeUtil
extends java.lang.Object

METHODS

- *build*
  `public Gui` **`build`**`( java.lang.String` **`filename`**`, java.lang.String` **`guiName`**`, com.imagos.javaglade.Gui` **`parentGui`**`, java.lang.String` **`rootnode`**`, Widget` **`parent`**`, java.lang.String` **`parentName`**`, java.lang.String` **`domain`**`, int` **`backgroundColor`** `)`
  - **Usage**
    - Creates a new JavaGlade (GladeXML equivalent) object (and the corresponding widgets) from the XML file fname. Optionally it will only build the interface from the widget node root (if it is not null). This feature is useful if you only want to build say a toolbar or menu from the XML file, but not the window it is embedded in. Note also that the XML parse tree is cached to speed up creating another JGlade object for the same file
  - **Parameters**

    - `rootnode` - String, the widget node in fname to start building from (or null)
    - `parent` - Widget, parent widget where the parsed interface will be embedded.
    - `parentName` - String, parent widget name.
    - `domain` - String, the translation domain for the XML file (or null for default)
    - `backgroundColor` - int, widget's background color.
  - **Exceptions**
    - `com.imagos.javaglade.JavaGladeException` - if construction fails

- *build*

  `public Gui` **build**`( java.lang.String` **filename**`, java.lang.String` **guiName**`, com.imagos.javaglade.Gui` **parentGui**`, com.imagos.javaglade.toolkit.swt.SwtTopLevelWindow` **parentWindow**`, int` **backgroundColor** `)`

  - **Usage**
    - Creates a new JavaGlade (GladeXML equivalent) object (and the corresponding widgets) from the XML file fname. This builder implements the creation of a JavaGlade object with no main window (as for example a set of dialogs). It gets the main window of another JavaGlade instance. Note also that the XML parse tree is cached to speed up creating another JGlade object for the same file
  - **Parameters**
    - `filename` - String, the XML (.glade) file path.
    - `parentGui` - Gui, GUI instance where the GUI that is being built will be embedded.
    - `parentWindow` - SwtTopLevelWindow, parent window for the created JavaGlade GUI contents.
    - `backgroundColor` - int, widget's background color.
  - **Exceptions**
    - `com.imagos.javaglade.JavaGladeException` - if construction fails

- *build*

  `public Gui` **build**`( java.lang.String` **filename**`, java.lang.String` **guiName**`, int` **backgroundColor** `)`

  - **Usage**
    - Creates a new JavaGlade (GladeXML equivalent) object (and the corresponding widgets) from the XML file fname.
  - **Parameters**
    - `filename` - String, the XML (.glade) file path.
    - `backgroundColor` - int, widget's background color.
  - **Exceptions**
    - `com.imagos.javaglade.JavaGladeException` - if construction fails

- *configure*

  `public void` **configure**`( com.imagos.javaglade.toolkit.swt.SwtToolkit` **swtToolkit**`, I18n` **i18n**`, java.lang.String` **resourceDir**`, com.imagos.javaglade.Localizer` **localizer** `)`

- *getColor*

  `public Color` **getColor**`( int` **color** `)`

    - Gets a color defined by the constant.
  - **Parameters**
    - `color` - int, color constant.
  - **Returns** - Color, color requested.

- *getDialog*
  `public SwtDialog` **getDialog**`( java.lang.String` **name** `)`
  - **Usage**
    - Gets the dialog widget from the widgets' catalog.
  - **Parameters**
    - `name` - String, widget name of the dialog to be searched.
  - **Returns** - SwtDialog, Dialog widget found (NULL in case no widget could be found).

- *getGladeInterface*
  `public GladeInterface` **getGladeInterface**`( java.lang.String` **filename** `)`
  - **Parameters**
    - `filename` - String, the XML (.glade) file path.
  - **Returns** - GladeInterface, parsed GladeInterface structure with the hierarchy of Gtk widgets
  - **Exceptions**
    - `com.imagos.javaglade.JavaGladeException` - if any error takes place during the access/construction of the selected glade GUI.

- *getI18n*
  `public I18n` **getI18n**`( )`

- *getInstance*
  `public static final JavaGladeUtil` **getInstance**`( )`
  - **Usage**
    - Returns the unique instance of the JavaGladeUtil class. Implements the singleton dessign pattern.
  - **Returns** - JavaGladeUtil, unique instance of JavaGladeUtil.

- *getLocalizer*
  `public Localizer` **getLocalizer**`( )`

- *getMsgCatalog*
  `public ResultMessageCatalog` **getMsgCatalog**`( )`

- *getResource*
  `public InputStream` **getResource**`( java.lang.String` **resourceName** `)`
  - **Usage**
    - Opens an InputStream to the resource file. It uses the external localizer instance to search for the resource.
  - **Parameters**
    - `resourceName` - String, file name of the resource file.

- *getResourceDir*
  `public String` **getResourceDir**`( )`

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getSwtToolkit*
  `public SwtToolkit` **getSwtToolkit**`( )`

- *getWidget*
  `public Widget` **getWidget**`( java.lang.String` **name** `)`
  - **Usage**
    - This function is used to get the Widget corresponding to the name in the interface description. You would use this if you have to do anything to the widget after loading.
  - **Parameters**
    - `name` - String, the name of the widget.
  - **Returns** - the widget matching name, or null if not exist

- *getWindow*
  `public SwtTopLevelWindow` **getWindow**`( java.lang.String` **name** `)`
  - **Usage**
    - This function is used to get the Window corresponding to the name in the interface description. You would use this if you have to do anything to the window after loading.
  - **Parameters**
    - `name` - String, the name of the window.
  - **Returns** - SwtTopLevelWindow, the window matching the name, or null if not exist.

- *initLogger*
  `public void` **initLogger**`( com.imagos.javaglade.Localizer` **loc**`, java.lang.String` **logDir**`, java.lang.String` **logConfigFile** `)`
  - **Usage**
    - Inits and configures the logger for the JavaGlade module.
  - **Parameters**
    - `logDir` - String, path to the log directory where the Javaglade log will be stored.
    - `logConfigFile` - String, path to the JavaGlade log config file.

- *setLocalizer*
  `public void` **setLocalizer**`( com.imagos.javaglade.Localizer` **loc** `)`

Chapter 3

Package com.imagos.javaglade.handlers

Package Contents *Page*

Classes

3.1 Classes

3.1.1 Class **GenericHandler**

Abstract class definition with the common methods defined for every Glade signal handler. Implements a method to link the handler to the JavaGlade instance used in the construction of the interface. That way the handler may have access to the GUI widgets and their data. <br/>Responsibilities:

- Define common methods and attributes for the Glade signal handlers.</li>

<br/>

Declaration

public abstract class GenericHandler
extends java.lang.Object

Constructors

- *GenericHandler*
  `public` **GenericHandler**`( )`

Methods

- *getGui*
  `public final Gui` **getGui**`( )`
  - **Usage**
    - Gets the Gui instance that contains all the handled widgets.
  - **Returns** - Gui, instance of Gui.
- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.
- *setGui*
  `public final void` **setGui**`( com.imagos.javaglade.Gui` **gui** `)`
  - **Usage**
    - Sets up the Gui instance that contains all the handled widgets.
  - **Parameters**
    - `gui` - Gui, instance of Gui.

3.1.2 Class **TestEventHandler**

DECLARATION

public class TestEventHandler
extends com.imagos.javaglade.handlers.GenericHandler

METHODS

- *getInstance*
  `public static final TestEventHandler` **getInstance**`( )`
- *on_button1_activate*
  `public void` **on_button1_activate**`( Event e )`
- *on_button1_clicked*
  `public void` **on_button1_clicked**`( Event e )`
- *on_button1_enter*
  `public void` **on_button1_enter**`( Event e )`
- *on_button1_leave*
  `public void` **on_button1_leave**`( Event e )`
- *on_button1_pressed*
  `public void` **on_button1_pressed**`( Event e )`
- *on_button1_released*
  `public void` **on_button1_released**`( Event e )`
- *on_button4_clicked*
  `public void` **on_button4_clicked**`( Event e )`
- *on_buttonhijo_pressed*
  `public void` **on_buttonhijo_pressed**`( Event e )`
  - **Usage**
    - Example of changing the panel contents using other glade files to implement the child (or internal) panel contents.
  - **Parameters**
    - `e` -
- *on_buttonhijo2_pressed*
  `public void` **on_buttonhijo2_pressed**`( Event e )`
  - **Usage**
    - Example of changing the panel contents using other glade files to implement the child (or internal) panel contents.
  - **Parameters**
    - `e` -
- *on_calendar1_day_selected_double_click*
  `public void` **on_calendar1_day_selected_double_click**`( Event e )`

- *on_calendar1_day_selected*
  `public void` **on_calendar1_day_selected**`( Event e )`

- *on_cancelbutton1_clicked*
  `public void` **on_cancelbutton1_clicked**`( Event e )`

- *on_combobox1_changed*
  `public void` **on_combobox1_changed**`( Event e )`

- *on_combobox1_check_resize*
  `public void` **on_combobox1_check_resize**`( Event e )`

- *on_combobox1_enter_notify_event*
  `public void` **on_combobox1_enter_notify_event**`( Event e )`

- *on_combobox1_leave_notify_event*
  `public void` **on_combobox1_leave_notify_event**`( Event e )`

- *on_label1_move_cursor*
  `public void` **on_label1_move_cursor**`( Event e )`

- *on_menuitem9_activate*
  `public void` **on_menuitem9_activate**`( Event e )`

- *on_nuevo3_activate*
  `public void` **on_nuevo3_activate**`( Event e )`

- *on_okbutton1_clicked*
  `public void` **on_okbutton1_clicked**`( Event e )`

- *on_radiobutton1_group_changed*
  `public void` **on_radiobutton1_group_changed**`( Event e )`

- *on_resultMessageText_modify*
  `public void` **on_resultMessageText_modify**`( Event e )`
  - **Usage**
    - Handler for the result message text modify event. Updates the scroll bars needed to see the result messages on screen.
  - **Parameters**
    - `e` - Event, result message text modify event data.

- *on_spinbutton1_change_value*
  `public void` **on_spinbutton1_change_value**`( Event e )`

- *on_spinbutton1_value_changed*
  `public void` **on_spinbutton1_value_changed**`( Event e )`

- *on_textview1_delete_from_cursor*
  `public void` **on_textview1_delete_from_cursor**`( Event e )`

- *on_textview1_insert_at_cursor*
  `public void` **on_textview1_insert_at_cursor**`( Event e )`

- *on_textview1_modify*
  `public void` **on_textview1_modify**`( Event e )`

- *on_textview1_move_cursor*
  `public void` **on_textview1_move_cursor**`( Event` **e** `)`

- *on_textview1_move_focus*
  `public void` **on_textview1_move_focus**`( Event` **e** `)`

- *on_togglebutton1_toggled*
  `public void` **on_togglebutton1_toggled**`( Event` **e** `)`

- *on_window1_check_resize*
  `public void` **on_window1_check_resize**`( Event` **e** `)`

METHODS INHERITED FROM CLASS `com.imagos.javaglade.handlers.GenericHandler`

(in 3.1.1, page 83)

- *getGui*
  `public final Gui` **getGui**`( )`
  - **Usage**
    - ∗ Gets the Gui instance that contains all the handled widgets.
  - **Returns** - Gui, instance of Gui.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - ∗ Returns tag Id assigned to CVS source file.

- *setGui*
  `public final void` **setGui**`( com.imagos.javaglade.Gui` **gui** `)`
  - **Usage**
    - ∗ Sets up the Gui instance that contains all the handled widgets.
  - **Parameters**
    - ∗ `gui` - Gui, instance of Gui.

Chapter 4

Package com.imagos.javaglade.parser

4.1 Classes

4.1.1 CLASS **GladeEntityResolver**

This class implements a SAX Entity Resolver to update the DTD used in the Glade XML Parsing to the local version. That way the application can work even with no Internet connection and the parsing is faster. <br/>Responsibilities:

- Translate the web URL of the DTD to a local URI.</li>

<br/>

DECLARATION

public class GladeEntityResolver
extends java.lang.Object
implements org.xml.sax.EntityResolver

CONSTRUCTORS

- *GladeEntityResolver*
  `public` **GladeEntityResolver**`( )`

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *resolveEntity*
  `public InputSource` **resolveEntity**`( java.lang.String` **publicId**, `java.lang.String` **systemId** `)`
  - **Usage**
    - Allow the application to resolve external entities. Its the implementation of the resolveEntity method of the EntityResolver interface. In this method the DTD external entity will be redirected to its local version.
  - **Parameters**
    - `publicId` - String, The public identifier of the external entity being referenced, or null if none was supplied.
    - `systemId` - String, The system identifier of the external entity being referenced.
  - **Returns** - InputSource An InputSource object describing the new input source, or null to request that the parser open a regular URI connection to the system identifier.

4.1.2 Class **GladeParser**

This class implements a SAX Parser to parse an XML Glade file. <br/>Responsibilities:

- Parse XML glade files.</li>
- Validate XML glade files.</li>

<br/>

Declaration

public class GladeParser
extends org.xml.sax.helpers.DefaultHandler

Constructors

- *GladeParser*
  `public` **GladeParser**`(` `java.io.InputStream` **is**, `I18n` **i18n**, `ResultMessageCatalog` **msgCatalog** `)`
  - **Usage**
    - Parser constructor. Creates a new instance of the parser and parses the file specified.
  - **Parameters**
    - `is` - InputStream, input stream of the glade file that will be parsed.
    - `i18n` - I18n, internationalization object to translate the texts into the current user's locale.
    - `msgCatalog` - ResultMessageCatalog, instance of the atalog of result messages defined for the JavaGlade module.

- *GladeParser*
  `public` **GladeParser**`(` `java.lang.String` **aPathFile**, `I18n` **i18n**, `ResultMessageCatalog` **msgCatalog** `)`
  - **Usage**
    - Parser constructor. Creates a new instance of the parser and parses the file specified.
  - **Parameters**
    - `aPathFile` - String, path of the glade file that will be parsed.
    - `i18n` - I18n, internationalization object to translate the texts
    - `msgCatalog` - ResultMessageCatalog, instance of the atalog of result messages defined for the JavaGlade module. into the current user's locale.

Methods

- *appendContent*
  `public void` **appendContent**`(` `char []` **ch**, `int` **start**, `int` **length** `)`

    - Updates the content of a PCDATA XML element with the parsed characters.
  - **Parameters**
    - `ch` - char[], set of characters that is being parsed.
    - `start` - int, starting index of the first character parsed.
    - `length` - int, number of characters parsed.

- *characters*
  `public void` **characters**`( char [] ` **ch**`, int ` **start**`, int ` **length** `)`
  - **Usage**
    - Receive notification of character data inside an element.
  - **Parameters**
    - `ch` - char[], The characters.
    - `start` - int, The start position in the character array.
    - `length` - int, The number of characters to use from the character array.
  - **Exceptions**
    - `SAXException,` - if there is a problem processing the glade file (such as incorrect file format, format not compliant with the schema, unknown tag appeared, etc.).

- *decWidgetDepth*
  `public void` **decWidgetDepth**`( )`
  - **Usage**
    - Decreases the number of widgets being parsed.

- *endElement*
  `public void` **endElement**`( java.lang.String ` **uri**`, java.lang.String` **localName**`, java.lang.String ` **qName** `)`
  - **Usage**
    - Receive notification of the end of an element.
  - **Parameters**
    - `uri` - String,
    - `localName` - String, the local name (without prefix), or the empty string if Namespace processing is not being performed.
    - `qName` - String, the qualified name (with prefix), or the empty string if qualified names are not available.
  - **Exceptions**
    - `SAXException,` - if there is a problem processing the glade file (such as incorrect file format, format not compliant with the schema, unknown tag appeared, etc.).

- *error*
  `public void` **error**`( org.xml.sax.SAXParseException ` **e** `)`
  - **Usage**
    - Receive notification of a recoverable parser error.
  - **Parameters**
    - `e` - SAXParseException, The warning information encoded as an exception.

- *fatalError*

  - **Usage**
    - Report a fatal XML parsing error.
  - **Parameters**
    - `e` - SAXParseException, The warning information encoded as an exception.

- *getContent*
  `public String` **getContent**`( )`
  - **Usage**
    - Gets the PCDATA XML element content.
  - **Returns** - String, parsed content.

- *getFilename*
  `public String` **getFilename**`( )`
  - **Usage**
    - Gets the path to the XML glade file.
  - **Returns** - String, glade file path.

- *getGladeAccelerator*
  `public GladeAccelerator` **getGladeAccelerator**`( )`
  - **Usage**
    - Gets the temporary accelerator data.
  - **Returns** - GladeAccelerator, temporary accelerator data.

- *getGladeAccessibility*
  `public GladeAccessibility` **getGladeAccessibility**`( )`
  - **Usage**
    - Gets the temporary accessibility data.
  - **Returns** - GladeAccessibility, temporary accessibility data.

- *getGladeChild*
  `public GladeChild` **getGladeChild**`( )`
  - **Usage**
    - Gets the temporary child data.
  - **Returns** - GladeChild, temporary child data.

- *getGladeInterface*
  `public GladeInterface` **getGladeInterface**`( )`
  - **Usage**
    - Gets the temporary glade interface data.
  - **Returns** - GladeInterface, temporary glade interface data.

- *getGladePacking*
  `public GladePacking` **getGladePacking**`( )`

    - Gets the temporary packing data.
  - **Returns** - GladePacking, temporary packing data.

- *getGladePlaceHolder*
  `public GladePlaceHolder` **getGladePlaceHolder**`( )`
  - **Usage**
    - Gets the temporary placeholder data.
  - **Returns** - GladePlaceHolder, temporary placeholder data.

- *getGladeProject*
  `public GladeProject` **getGladeProject**`( )`
  - **Usage**
    - Gets the temporary project data.
  - **Returns** - GladeProject, temporary project data.

- *getGladeProperty*
  `public GladeProperty` **getGladeProperty**`( )`
  - **Usage**
    - Gets the temporary property data.
  - **Returns** - GladeProperty, temporary property data.

- *getGladeSignal*
  `public GladeSignal` **getGladeSignal**`( )`
  - **Usage**
    - Gets the temporary signal data.
  - **Returns** - GladeSignal, temporary signal data.

- *getGladeWidget*
  `public GladeWidget` **getGladeWidget**`( )`
  - **Usage**
    - Gets the temporary widget data.
  - **Returns** - GladeWidget, temporary widget data.

- *getPropType*
  `public GladeParser.PropertyType` **getPropType**`( )`
  - **Usage**
    - Gets the property type of the properties section that is being parsed.
  - **Returns** - PropertyType, property type being parsed.
  - **See Also**
    - `com.imagos.javaglade.parser.com.imagos.javaglade.parser.PropertyType`

- *getRevision*
  `public static String` **getRevision**`( )`

    - Returns tag Id assigned to CVS source file.

- *getSChildren*
  `public Stack` **getSChildren**`( )`
  - **Usage**
    - Gets the stack of children being parsed at the time.
  - **Returns** - Stack<GladeChild>, stack of children being parsed.

- *getSWidgets*
  `public Stack` **getSWidgets**`( )`
  - **Usage**
    - Gets the stack of widgets being parsed at the time.
  - **Returns** - Stack<GladeWidget>, stack of widgets being parsed.

- *getWidgetDepth*
  `public int` **getWidgetDepth**`( )`
  - **Usage**
    - Gets the number of widgets being parsed at the time.
  - **Returns** - int, number of widgets being parsed.

- *incWidgetDepth*
  `public void` **incWidgetDepth**`( )`
  - **Usage**
    - Increases the number of widgets being parsed.

- *setContent*
  `public void` **setContent**`( java.lang.String` **content** `)`
  - **Usage**
    - Sets up the PCDATA XML element content.
  - **Parameters**
    - `content` - String, parsed content.

- *setFilename*
  `public void` **setFilename**`( java.lang.String` **filename** `)`
  - **Usage**
    - Sets up the path to the XML glade file.
  - **Parameters**
    - `filename` - String, glade file path.

- *setGladeAccelerator*
  `public void` **setGladeAccelerator**`( com.imagos.javaglade.vo.GladeAccelerator` **gladeAccelerator** `)`
  - **Usage**
    - Sets up the temporary accelerator data.

    - `gladeAccelerator` - GladeAccelerator, temporary accelerator data.

- *setGladeAccessibility*
  `public void` **setGladeAccessibility**`( com.imagos.javaglade.vo.GladeAccessibility` **gladeAccessibility** `)`
  - **Usage**
    - Sets up the temporary accessibility data.
  - **Parameters**
    - `gladeAccessibility` - GladeAccessibility, temporary accessibility data.

- *setGladeChild*
  `public void` **setGladeChild**`( com.imagos.javaglade.vo.GladeChild` **gladeChild** `)`
  - **Usage**
    - Sets up the temporary child data.
  - **Parameters**
    - `gladeChild` - GladeChild, temporary child data.

- *setGladeInterface*
  `public void` **setGladeInterface**`( com.imagos.javaglade.vo.GladeInterface` **gladeInterface** `)`
  - **Usage**
    - Sets up the temporary glade interface data.
  - **Parameters**
    - `gladeInterface` - GladeInterface, temporary glade interface data.

- *setGladePacking*
  `public void` **setGladePacking**`( com.imagos.javaglade.vo.GladePacking` **gladePacking** `)`
  - **Usage**
    - Sets up the temporary packing data.
  - **Parameters**
    - `gladePacking` - GladePacking, temporary packing data.

- *setGladePlaceHolder*
  `public void` **setGladePlaceHolder**`( com.imagos.javaglade.vo.GladePlaceHolder` **gladePlaceHolder** `)`
  - **Usage**
    - Sets up the temporary placeholder data.
  - **Parameters**
    - `gladePlaceHolder` - GladePlaceHolder, temporary placeholder data.

- *setGladeProject*
  `public void` **setGladeProject**`( com.imagos.javaglade.vo.GladeProject` **gladeProject** `)`

    - Sets up the temporary project data.
  - **Parameters**
    - `gladeProject` - GladeProject, temporary project data.

- *setGladeProperty*
  `public void` **setGladeProperty**`( com.imagos.javaglade.vo.GladeProperty` **gladeProperty** `)`
  - **Usage**
    - Sets up the temporary property data.
  - **Parameters**
    - `gladeProperty` - GladeProperty, temporary property data.

- *setGladeSignal*
  `public void` **setGladeSignal**`( com.imagos.javaglade.vo.GladeSignal` **gladeSignal** `)`
  - **Usage**
    - Sets up the temporary signal data.
  - **Parameters**
    - `gladeSignal` - GladeSignal, temporary signal data.

- *setGladeWidget*
  `public void` **setGladeWidget**`( com.imagos.javaglade.vo.GladeWidget` **gladeWidget** `)`
  - **Usage**
    - Sets up the temporary widget data.
  - **Parameters**
    - `gladeWidget` - GladeWidget, temporary widget data.

- *setPropType*
  `public void` **setPropType**`( com.imagos.javaglade.parser.GladeParser.PropertyType` **propType** `)`
  - **Usage**
    - Sets up the property type of the properties section that is being parsed.
  - **Parameters**
    - `propType` - PropertyType, property type being parsed.
  - **See Also**
    - `com.imagos.javaglade.parser.com.imagos.javaglade.parser.PropertyType`

- *setSChildren*
  `public void` **setSChildren**`( java.util.Stack` **children** `)`
  - **Usage**
    - Sets up the stack of children being parsed at the time.
  - **Parameters**
    - `children` - Stack<GladeChild>, stack of children being parsed.

- *setSWidgets*
  `public void` **setSWidgets**`( java.util.Stack` **widgets** `)`
  - **Usage**
    - Sets up the stack of widgets being parsed at the time.
  - **Parameters**
    - `widgets` - Stack<GladeWidget>, stack of widgets being parsed.

- *setWidgetDepth*
  `public void` **setWidgetDepth**`( int` **widgetDepth** `)`
  - **Usage**
    - Sets up the number of widgets being parsed at the time.
  - **Parameters**
    - `widgetDepth` - int, number of widgets being parsed.

- *startDocument*
  `public void` **startDocument**`( )`
  - **Usage**
    - Receive notification of the start of the glade file to parse.
  - **Exceptions**
    - `SAXException,` - if there is a problem processing the glade file (such as incorrect file format, format not compliant with the schema, unknown tag appeared, etc.).

- *startElement*
  `public void` **startElement**`( java.lang.String` **uri**`, java.lang.String` **localName**`, java.lang.String` **qName**`, org.xml.sax.Attributes` **attributes** `)`
  - **Usage**
    - Receive notification of the start of an element.
  - **Parameters**
    - `uri` - String,
    - `localName` - String, the local name (without prefix), or the empty string if Namespace processing is not being performed.
    - `qName` - String, the qualified name (with prefix), or the empty string if qualified names are not available.
    - `attributes` - Attributes, the specified or defaulted attributes.
  - **Exceptions**
    - `SAXException,` - if there is a problem processing the glade file (such as incorrect file format, format not compliant with the schema, unknown tag appeared, etc.).

- *warning*
  `public void` **warning**`( org.xml.sax.SAXParseException` **e** `)`
  - **Usage**
    - Receive notification of a parser warning.
  - **Parameters**
    - `e` - SAXParseException, The warning information encoded as an exception.

METHODS INHERITED FROM CLASS `org.xml.sax.helpers.DefaultHandler`

- *characters*
  `public void` **characters**`( char [] ` **arg0**`, int ` **arg1**`, int ` **arg2** `)`
- *endDocument*
  `public void` **endDocument**`( )`
- *endElement*
  `public void` **endElement**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1**`, java.lang.String ` **arg2** `)`
- *endPrefixMapping*
  `public void` **endPrefixMapping**`( java.lang.String ` **arg0** `)`
- *error*
  `public void` **error**`( org.xml.sax.SAXParseException ` **arg0** `)`
- *fatalError*
  `public void` **fatalError**`( org.xml.sax.SAXParseException ` **arg0** `)`
- *ignorableWhitespace*
  `public void` **ignorableWhitespace**`( char [] ` **arg0**`, int ` **arg1**`, int ` **arg2** `)`
- *notationDecl*
  `public void` **notationDecl**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1**`, java.lang.String ` **arg2** `)`
- *processingInstruction*
  `public void` **processingInstruction**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1** `)`
- *resolveEntity*
  `public InputSource` **resolveEntity**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1** `)`
- *setDocumentLocator*
  `public void` **setDocumentLocator**`( org.xml.sax.Locator ` **arg0** `)`
- *skippedEntity*
  `public void` **skippedEntity**`( java.lang.String ` **arg0** `)`
- *startDocument*
  `public void` **startDocument**`( )`
- *startElement*
  `public void` **startElement**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1**`, java.lang.String ` **arg2**`, org.xml.sax.Attributes ` **arg3** `)`
- *startPrefixMapping*
  `public void` **startPrefixMapping**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1** `)`
- *unparsedEntityDecl*
  `public void` **unparsedEntityDecl**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1**`, java.lang.String ` **arg2**`, java.lang.String ` **arg3** `)`
- *warning*
  `public void` **warning**`( org.xml.sax.SAXParseException ` **arg0** `)`

4.1.3 CLASS **GladeParser.PropertyType**

Types of properties to parse. There can be these types:

- Property of a <widget>item. </li>
- Property of a item. </li>
- Property of a <packing>item. </li>

DECLARATION

public static final class GladeParser.PropertyType
extends java.lang.Enum

FIELDS

- public static final GladeParser.PropertyType PROPERTY_NONE
  -
- public static final GladeParser.PropertyType PROPERTY_WIDGET
  -
- public static final GladeParser.PropertyType PROPERTY_ATK
  -
- public static final GladeParser.PropertyType PROPERTY_PACKING
  -

METHODS

- *valueOf*
  `public static GladeParser.PropertyType` **valueOf**`( java.lang.String` **name** `)`
- *values*
  `public static GladeParser.PropertyType` **values**`( )`

METHODS INHERITED FROM CLASS `java.lang.Enum`

- *clone*
  `protected final Object` **clone**`( )`
- *compareTo*
  `public final int` **compareTo**`( java.lang.Enum` **arg0** `)`
- *equals*
  `public final boolean` **equals**`( java.lang.Object` **arg0** `)`
- *finalize*
  `protected final void` **finalize**`( )`
- *getDeclaringClass*
  `public final Class` **getDeclaringClass**`( )`
- *hashCode*
  `public final int` **hashCode**`( )`
- *name*
  `public final String` **name**`( )`
- *ordinal*
  `public final int` **ordinal**`( )`
- *toString*
  `public String` **toString**`( )`
- *valueOf*

Chapter 5

Package com.imagos.javaglade.toolkit.swt

Package Contents *Page*

Classes

5.1 Classes

5.1.1 CLASS **ButtonBoxLayout**

This class implements a compliant layout for the GtkHButtonBox widget. This widget can lay out the components in four different styles: EDGE, SPREAD, START and END. All of them are implemented here in this custom layout. <br/>

- EDGE: Lays out the first widget at the top left square and the last at the top right square. The rest of components are evenly layed out in the rest of available space with even separators.</li>
- SPREAD: Lays out the widgets separated by even free spaces.</li>
- START: Lays out the widgets without separation at the left side.</li>
- END: Lays out the widgets without separation at the right side.</li>

<br/>Responsibilities:

- Provide compliant layout for the GTK HButtonBox widget layout style.</li>

<br/>

DECLARATION

public class ButtonBoxLayout
extends Layout

FIELDS

- public static final int LAYOUT_STYLE_EDGE
  - Constant for the layout style EDGE.
- public static final int LAYOUT_STYLE_SPREAD
  - Constant for the layout style SPREAD.
- public static final int LAYOUT_STYLE_START
  - Constant for the layout style START.
- public static final int LAYOUT_STYLE_END
  - Constant for the layout style END.

CONSTRUCTORS

- *ButtonBoxLayout*
  `public` **ButtonBoxLayout**`( int` **style** `)`
  - **Usage**
    - Default constructor.
  - **Parameters**
    - `style` - int, layout style selected.

    - `com.imagos.javaglade.toolkit.swt.ButtonBoxLayout.LAYOUT_STYLE_EDGE`
    - `com.imagos.javaglade.toolkit.swt.ButtonBoxLayout.LAYOUT_STYLE_SPREAD`
    - `com.imagos.javaglade.toolkit.swt.ButtonBoxLayout.LAYOUT_STYLE_START`
    - `com.imagos.javaglade.toolkit.swt.ButtonBoxLayout.LAYOUT_STYLE_END`

METHODS

- *computeSize*
  `protected Point` **computeSize**`( Composite` **composite**`, int` **wHint**`, int` **hHint**`, boolean` **changed** `)`
  - **Usage**
    - Computes and returns the size of the specified composite's client area according to this layout. <br/>This method computes the size that the client area of the composite must be in order to position all children at their preferred size inside the composite according to the layout algorithm encoded by this layout.
      When a width or height hint is supplied, it is used to constrain the result. For example, if a width hint is provided that is less than the width of the client area, the layout may choose to wrap and increase height, clip, overlap, or otherwise constrain the children.
  - **Parameters**
    - `composite` - Composite, a composite widget using this layout.
    - `wHint` - int, width (SWT.DEFAULT for preferred size).
    - `hHint` - int, height (SWT.DEFAULT for preferred size).
    - `changed` - boolean, True means flush cached layout values.
  - **Returns** - Point, a point containing the computed size (width, height).

- *getMarginHeight*
  `public int` **getMarginHeight**`( )`
  - **Usage**
    - Gets the value of the vertical margin.
  - **Returns** - int, vertical margin.

- *getMarginWidth*
  `public int` **getMarginWidth**`( )`
  - **Usage**
    - Gets the value of the horizontal margin.
  - **Returns** - int, horizontal margin.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getSeparation*

- **Usage**
  - Gets the value of the horizontal separation.
- **Returns** - int, horizontal separation.

- *getStyle*
  `public int` **getStyle**`( )`
  - **Usage**
    - Gets the style of the ButtonBoxLayout.
  - **Returns** - int, ButtonBoxLayout style.

- *layout*
  `protected void` **layout**`( Composite` **composite**`, boolean` **changed** `)`
  - **Usage**
    - Lays out the children of the specified composite according to this layout. <br/>This method positions and sizes the children of a composite using the layout algorithm encoded by this layout. Children of the composite are positioned in the client area of the composite. The position of the composite is not altered by this method. <br/>When the flush cache hint is true, the layout is instructed to flush any cached values associated with the children. Typically, a layout will cache the preferred sizes of the children to avoid the expense of computing these values each time the widget is laid out. <br/>When layout is triggered explicitly by the programmer the flush cache hint is true. When layout is triggered by a resize, either caused by the programmer or by the user, the hint is false. <br/>
  - **Parameters**
    - `composite` - Composite, a composite widget using this layout.
    - `changed` - boolean, True means flush cached layout values

- *setMarginHeight*
  `public void` **setMarginHeight**`( int` **marginHeight** `)`
  - **Usage**
    - Sets up the value of the vertical margin.
  - **Parameters**
    - `marginHeight` - int, vertical margin.

- *setMarginWidth*
  `public void` **setMarginWidth**`( int` **marginWidth** `)`
  - **Usage**
    - Sets up the value of the horizontal margin.
  - **Parameters**
    - `marginWidth` - int, horizontal margin.

- *setSeparation*
  `public void` **setSeparation**`( int` **separation** `)`
  - **Usage**
    - Sets up the value of the horizontal separation.
  - **Parameters**

METHODS INHERITED FROM CLASS `Layout`

5.1.2 CLASS **HorizontalData**

DECLARATION

public class HorizontalData
extends java.lang.Object

CONSTRUCTORS

- *HorizontalData*
  `public` **HorizontalData**`( )`
- *HorizontalData*
  `public` **HorizontalData**`( boolean` **filling** `)`

METHODS

- *getVerticalAlignment*
  `public int` **getVerticalAlignment**`( )`
- *isExpanding*
  `public boolean` **isExpanding**`( )`
- *isFilling*
  `public boolean` **isFilling**`( )`
- *setExpanding*
  `public void` **setExpanding**`( boolean` **expanding** `)`
- *setFilling*
  `public void` **setFilling**`( boolean` **filling** `)`
- *setVerticalAlignment*
  `public void` **setVerticalAlignment**`( int` **verticalAlignment** `)`

5.1.3 CLASS **HorizontalLayout**

This class implements a compliant layout for the HBox widget. This widget lay out the components in an horizontal row and allows horizontal filling. <br/>Responsibilities:

- Provide compliant layout for the GTK HBox widget layout style.</li>

<br/>

DECLARATION

public class HorizontalLayout
extends Layout

CONSTRUCTORS

- *HorizontalLayout*
  `public` **HorizontalLayout**`( )`
  - **Usage**
    - Default constructor.

- *HorizontalLayout*
  `public` **HorizontalLayout**`( int` **separation** `)`
  - **Usage**
    - Default constructor.

METHODS

- *computeSize*
  `protected Point` **computeSize**`( Composite` **composite**`, int` **wHint**`, int` **hHint**`, boolean` **changed** `)`
  - **Usage**
    - Computes and returns the size of the specified composite's client area according to this layout. <br/>This method computes the size that the client area of the composite must be in order to position all children at their preferred size inside the composite according to the layout algorithm encoded by this layout.
      When a width or height hint is supplied, it is used to constrain the result. For example, if a width hint is provided that is less than the width of the client area, the layout may choose to wrap and increase height, clip, overlap, or otherwise constrain the children.
  - **Parameters**
    - `composite` - Composite, a composite widget using this layout.
    - `wHint` - int, width (SWT.DEFAULT for preferred size).
    - `hHint` - int, height (SWT.DEFAULT for preferred size).
    - `changed` - boolean, True means flush cached layout values.
  - **Returns** - Point, a point containing the computed size (width, height).

- *getMarginHeight*
  `public int` **getMarginHeight**`( )`
  - **Usage**
    - Gets the value of the vertical margin.
  - **Returns** - int, vertical margin.

- *getMarginWidth*
  `public int` **getMarginWidth**`( )`
  - **Usage**
    - Gets the value of the horizontal margin.
  - **Returns** - int, horizontal margin.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getSeparation*
  `public int` **getSeparation**`( )`
  - **Usage**
    - Gets the value of the horizontal separation width among children widgets.
  - **Returns** - int, horizontal separation in pixels.

- *layout*
  `protected void` **layout**`( Composite` **composite**`, boolean` **changed** `)`
  - **Usage**
    - Lays out the children of the specified composite according to this layout. <br/>This method positions and sizes the children of a composite using the layout algorithm encoded by this layout. Children of the composite are positioned in the client area of the composite. The position of the composite is not altered by this method. <br/>When the flush cache hint is true, the layout is instructed to flush any cached values associated with the children. Typically, a layout will cache the preferred sizes of the children to avoid the expense of computing these values each time the widget is laid out. <br/>When layout is triggered explicitly by the programmer the flush cache hint is true. When layout is triggered by a resize, either caused by the programmer or by the user, the hint is false. <br/>
  - **Parameters**
    - `composite` - Composite, a composite widget using this layout.
    - `changed` - boolean, True means flush cached layout values

- *setMarginHeight*
  `public void` **setMarginHeight**`( int` **marginHeight** `)`
  - **Usage**
    - Sets up the value of the vertical margin.
  - **Parameters**
    - `marginHeight` - int, vertical margin.

- *setMarginWidth*
  `public void` **setMarginWidth**`( int` **marginWidth** `)`
  - **Usage**

  - **Parameters**
    - `marginWidth` - int, horizontal margin.

- *setSeparation*
  `public void` **`setSeparation`**`( int` **`separation`** `)`
  - **Usage**
    - Sets up the value of the horizontal separation width among children widgets.
  - **Parameters**
    - `separation` - int, horizontal separation in pixels.

METHODS INHERITED FROM CLASS `Layout`

5.1.4 CLASS **SizeGrip**

A non-native size grip which looks (and almost feels) like the native Win32 size grip.

The SHADOW_IN style causes highlight lines to be drawn at the right and bottom border. This style should be used when placing the size grip on top of the bottom right corner of a FramedComposite with style SHADOW_IN. If the FLAT style is set, the size grip is drawn in a Windows XP style instead of the normal Windows Classic style.

Styles:</dt> SHADOW_IN, FLAT</dd>

Events:</dt> (none)</dd>

NOTE: The visibility of this widget is controlled by the "maximized" state of the shell. The size grip is hidden when the shell is maximized, even if it has been made visible by calling `setVisible(true)`. getVisible() always returns the value set with setVisible(). isVisible() returns the true visibility, as usual.

New in 1.6: Smoother resizing for a more native look & feel. The window size is not updated more than once every 25ms to reduce the number of unnecessary repaints.

New in 1.8: You can specify a parent Composite other than the shell which will be resized by the SizeGrip.

DECLARATION

public final class SizeGrip
extends Canvas

CONSTRUCTORS

- *SizeGrip*
  `public` **SizeGrip**`(` `Composite` **parent**`,` `Composite` **resizableParent**`,` `int` **style** `)`
- *SizeGrip*
  `public` **SizeGrip**`(` `Composite` **parent**`,` `int` **style** `)`

METHODS

- *computeSize*
  `public Point` **computeSize**`(` `int` **wHint**`,` `int` **hHint**`,` `boolean` **changed** `)`
- *getVisible*
  `public boolean` **getVisible**`( )`
- *isReparentable*
  `public boolean` **isReparentable**`( )`
- *setFocus*
  `public boolean` **setFocus**`( )`
- *setMinimumShellSize*
  `public void` **setMinimumShellSize**`(` `int` **width**`,` `int` **height** `)`
  - **Usage**
    - Set the allowed minimum size for the shell. The SizeGrip will not resize the shell to a smaller size.

      Note: This does <em>not</em>affect other ways of resizing the shell, like using the size controls which are placed on the trimmings by the window manager.
- *setMinimumShellSize*
  `public void` **setMinimumShellSize**`(` `Point` **p** `)`
  - **Usage**
    - Set the allowed minimum size for the shell. The SizeGrip will not resize the shell to a smaller size.

      Note: This does <em>not</em>affect other ways of resizing the shell, like using the size controls which are placed on the trimmings by the window manager.
- *setVisible*
  `public void` **setVisible**`(` `boolean` **visible** `)`

METHODS INHERITED FROM CLASS `Canvas`

5.1.5 Class **StatusBar**

Implementation of the GUI widget StatusBar for SWT. This widget is not directly supported for the SWT toolkit, so a custom widget must be created. The custom status bar widget will render both an status icon and an status label for showing the status text to the user. <br/>Responsibilities:

- Build a GUI widget representing a Statusbar in SWT.</li>
- Give accessors for updating and getting the status info showed to the user.</li>

<br/>

Declaration

public class StatusBar
extends Composite

Constructors

- *StatusBar*
  `public` **StatusBar**`( Composite` **parent** `)`
  - **Usage**
    - Default StatusBar constructor.
  - **Parameters**
    - `parent` - Composite, parent widget of the status bar.

- *StatusBar*
  `public` **StatusBar**`( Composite` **parent**`, int` **style** `)`
  - **Usage**
    - StatusBar constructor that initializes the style of the statusbar widget.
  - **Parameters**
    - `parent` - Composite, parent widget of the status bar.
    - `style` - int, SWT style constant.

Methods

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *updateStatus*
  `public void` **updateStatus**`( Image` **image** `)`
  - **Usage**
    - Updates the status icon showed to the user.

    - `image` - Image, new status icon.

- *updateStatus*
  `public void` **`updateStatus`**`( java.lang.String` **`text`** `)`
  - **Usage**
    - Update the status text showed to the user.
  - **Parameters**
    - `text` - String, new status text.

Methods inherited from class `Composite`

5.1.6 Class **SwtDialog**

Implementation of the GUI widget Dialog for SWT. This widget is an abstract definition of a dialog in the SWT Toolkit. It makes use of the several dialog widgets implemented by the SWT library, such as Dialog, FileDialog, FontDialog, etc. <br/>Responsibilities:

- Provide a generic class for encapsulating any SWT widget that may act as a dialog window. In Gtk a Dialog always hang from a gtk widget (usually composite widgets). In SWT, on the contrary, Dialog are linked directly to a shell (normally hang from the main window shell).</li>
- Provide Dialog typing.</li>
- Provide methods to build and open a dialog.</li>

<br/>

Declaration

public class SwtDialog
extends java.lang.Object

Fields

- public static final int FILECHOOSER_DIALOG_TYPE
  - Constant definition with the dialog type FILECHOOSER. It's the dialog type for open/save file selection dialogs.
- public static final int COLORSELECTION_DIALOG_TYPE
  - Constant definition with the dialog type COLORSELECTION. It's the dialog type for colour selection dialogs.
- public static final int FONTSELECTION_DIALOG_TYPE
  - Constant definition with the dialog type FONTSELECTION. It's the dialog type for font selection dialogs.
- public static final int DIRECTORYSELECTION_DIALOG_TYPE

  - Constant definition with the dialog type DIRECTORYSELECTION. It's the dialog type for directory selection dialogs.

- public static final int MESSAGEBOX_DIALOG_TYPE
  - Constant definition with the dialog type MESSAGEBOX. It's the dialog type for dialogs that show general text or information messages.
- public static final int DIALOG_TYPE
  - Constant definition with the dialog type DIALOG. It's the general type for custom dialogs. It doesn't correspond to any concrete type and the content of the dialog should be specified in the glade file.

CONSTRUCTORS

- *SwtDialog*
  `public` **SwtDialog**`( )`
  - **Usage**
    - Default SwtDialog constructor.

- *SwtDialog*
  `public` **SwtDialog**`( Shell` **parent**`, int` **style**`, int` **type**`, boolean` **isVisible** `)`
  - **Usage**
    - SwtDialog constructor that initializes the Dialog values.
  - **Parameters**
    - `parent` - Shell, parent window of the dialog (normally the application's main window).
    - `style` - int, SWT style constant for styling the dialog.
    - `type` - int, dialog's type.
    - `isVisible` - boolean, value for the isVisible flag. True if dialog must be showed at the starting of the application; False if the dialog will be showed when it's required.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FILECHOOSER_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.COLORSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FONTSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIRECTORYSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.MESSAGEBOX_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIALOG_TYPE`

METHODS

- *addDisposeListener*
  `public void` **addDisposeListener**`( DisposeListener` **listener** `)`

- *addListener*

- *close*
  `public void` **close**`( )`
  - **Usage**
    - Hides the dialog and shows again the parent window. This method doesn't dispose the dialog contents so the dialog can be reused when it must be showed again without rebuilding.
- *dispose*
  `public void` **dispose**`( )`
  - **Usage**
    - Removes the dialog and clears all the contents inside. Contrary to the close() method, this method disposes all the contents of the dialog so when the dialog is reopened, it must be previously rebuilt.
- *getButtonArea*
  `public Composite` **getButtonArea**`( )`
  - **Usage**
    - Gets the button panel where custom dialog button will be placed.
  - **Returns** - Composite, button composite panel of the dialog.
- *getContentArea*
  `public Composite` **getContentArea**`( )`
  - **Usage**
    - Gets the content panel where custom dialog widgets will be placed.
  - **Returns** - Composite, content composite panel of the dialog.
- *getDialog*
  `public Dialog` **getDialog**`( )`
  - **Usage**
    - Gets the SWT dialog widget that renders the SwtDialog.
  - **Returns** - Dialog, rendered SWT Dialog widget.
- *getResult*
  `public Object` **getResult**`( )`
  - **Usage**
    - Gets the result from the user got as the result of showing the dialog.
  - **Returns** - Object, result got from the dialog.
- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.
- *getShell*

  - **Usage**
    - Gets the shell (window) the dialog is linked to.
  - **Returns** - Shell, parent window of the dialog.

- *getType*
  `public int getType( )`
  - **Usage**
    - Gets the dialog's type.
  - **Returns** - int, dialog's type.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FILECHOOSER_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.COLORSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FONTSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIRECTORYSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.MESSAGEBOX_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIALOG_TYPE`

- *isVisible*
  `public boolean isVisible( )`
  - **Usage**
    - Gets the value of the isVisible flag that shows if the dialog is being showed to the user or not.
  - **Returns** - boolean, True if the dialog is showed to the user; False in any other case.

- *open*
  `public Object open( )`
  - **Usage**
    - Makes the dialog visible.
  - **Returns** - Object, response got from the user as the result of showing the dialog (for example filepath selected in a filechooser dialog, or colour selected in a colourselection dialog, etc.)

- *setButtonArea*
  `public void setButtonArea( Composite buttonArea )`
  - **Usage**
    - Sets up the button panel where custom dialog button will be placed.
  - **Parameters**
    - `buttonArea` - Composite, button composite panel of the dialog.

- *setContentArea*
  `public void setContentArea( Composite contentArea )`
  - **Usage**
    - Sets up the content panel where custom dialog widgets will be placed.

    - `contentArea` - Composite, content composite panel of the dialog.

- *setDialog*
  `public void` **setDialog**`(` `Dialog` **dialog** `)`
  - **Usage**
    - Sets up the SWT dialog widget that renders the SwtDialog.
  - **Parameters**
    - `dialog` - Dialog, rendered SWT Dialog widget.
- *setResult*
  `public void` **setResult**`(` `java.lang.Object` **result** `)`
  - **Usage**
    - Sets up the result from the user got as the result of showing the dialog.
  - **Parameters**
    - `result` - Object, result got from the dialog.
- *setShell*
  `public void` **setShell**`(` `Shell` **shell** `)`
  - **Usage**
    - Sets up the shell (window) the dialog is linked to.
  - **Parameters**
    - `shell` - Shell, parent window of the dialog.
- *setType*
  `public void` **setType**`(` `int` **type** `)`
  - **Usage**
    - Sets up the dialog's type.
  - **Parameters**
    - `type` - int, dialog's type.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FILECHOOSER_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.COLORSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.FONTSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIRECTORYSELECTION_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.MESSAGEBOX_DIALOG_TYPE`
    - `com.imagos.javaglade.toolkit.swt.SwtDialog.DIALOG_TYPE`
- *setVisible*
  `public void` **setVisible**`(` `boolean` **isVisible** `)`
  - **Usage**
    - Sets up the value of the isVisible flag that shows if the dialog is being showed to the user or not.
  - **Parameters**
    - `isVisible` - boolean, True if the dialog is showed to the user; False in any other

5.1.7 CLASS **SwtDialogCloseListener**

DECLARATION

public class SwtDialogCloseListener
extends java.lang.Object

CONSTRUCTORS

- *SwtDialogCloseListener*
  `public` **SwtDialogCloseListener**`( com.imagos.javaglade.toolkit.swt.SwtDialog` **dialog** `)`

METHODS

- *getDialog*
  `public SwtDialog` **getDialog**`( )`
- *handleEvent*
  `public void` **handleEvent**`( Event` **e** `)`
- *setDialog*
  `public void` **setDialog**`( com.imagos.javaglade.toolkit.swt.SwtDialog` **dialog** `)`

5.1.8 CLASS **SwtToolkit**

This class implements a SWT toolkit for generating the widgets. It's used in the conversion from Glade Gtk widgets into a corresponding SWT GUI. It will be in charge of the widget generation, initialization and configuration. <br/>Responsibilities:

- Provide widget generation methods for all supported widgets.</li>
- Provide suitable conversion from Gtk widget properties into SWT compliant values.</li>
- Provide suitable conversion from Gtk signals model into an SWT compliant event model.</li>
- Register the event handlers for the generated widgets.</li>

<br/>

DECLARATION

public class SwtToolkit
extends java.lang.Object

CONSTRUCTORS

- *SwtToolkit*
  `public` **SwtToolkit**`(` `java.lang.String` **resourceDir** `)`
  - **Usage**
    - Constructor of the SwtToolkit class
  - **Parameters**
    - `resourceDir` - String, local path to the resources used by the interface.

- *SwtToolkit*
  `public` **SwtToolkit**`(` `java.lang.String` **resourceDir**`,` `int` **backgroundColor** `)`
  - **Usage**
    - Constructor of the SwtToolkit class
  - **Parameters**
    - `resourceDir` - String, local path to the resources used by the interface.
    - `backgroundColor` - int, widget's background color.

METHODS

- *addTabItemContent*
  `public void` **addTabItemContent**`(` `Widget` **tabItem**`,` `Widget` **content** `)`
  - **Usage**
    - Sets up the content of a SWT tabItem widget. The content must be a valid SWT Control.
  - **Parameters**
    - `tabItem` - Widget, SWT tabItem widget.
    - `content` - Widget, tabItem content's widget.
  - **See Also**
    - `org.eclipse.swt.widgets.Control`

- *addToolItemContent*
  `public void` **addToolItemContent**`(` `Widget` **toolItem**`,` `Widget` **content** `)`
  - **Usage**
    - Sets up the content of a SWT toolItem widget. The content must be a valid SWT Control.
  - **Parameters**
    - `toolItem` - Widget, SWT toolItem widget.
    - `content` - Widget, toolItem content's widget.
  - **See Also**
    - `org.eclipse.swt.widgets.Control`

- *button*
  `public Widget` **button**`(` `Composite` **parent**`,` `java.lang.String` **name**`,`
  `java.util.ArrayList` **lProperties**`,` `com.imagos.javaglade.vo.GladePacking`

  - **Usage**
    - Build a widget of type PUSHBUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *button*
  `public Widget` **button**`( Composite` **parent**`, java.lang.String` **name**`, java.lang.String` **label** `)`
  - **Usage**
    - Build a widget of type PUSH_BUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `label` - String, label of the button widget.
  - **Returns** - Widget, the widget built.

- *calendar*
  `public Widget` **calendar**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type CALENDAR.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *checkButton*
  `public Widget` **checkButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type CHECKBUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *comboBox*
  `public Widget` **comboBox**`(` `Composite` **parent**`,` `java.lang.String` **name**`,` `java.util.ArrayList` **lProperties**`,` `com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type COMBOBOX.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *connectSignal*
  `public void` **connectSignal**`(` `java.lang.String` **signalName**`,` `Widget` **widget**`,` `java.lang.String` **handler**`,` `java.lang.Object` **objHandler** `)`
  - **Usage**
    - Connect a signal (event) to its correspondent event handler.
  - **Parameters**
    - `signalName` - String, SWT name of the signal (event type).
    - `widget` - Widget, SWT widget in which the events take place.
    - `handler` - String, handler method's name.
    - `objHandler` - Object, instance of the class that implements the handler.

- *coolBar*
  `public Widget` **coolBar**`(` `Composite` **parent**`,` `java.lang.String` **name**`,` `java.util.ArrayList` **lProperties**`,` `com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type COOL BAR (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *createColorSelectionDialog*
  `public SwtDialog` **createColorSelectionDialog**`(` `Shell` **shell**`,` `java.lang.String` **name**`,` `java.util.ArrayList` **lProperties** `)`
  - **Usage**
    - Creates a new Dialog of type COLORSELECTION. This dialog allows the user to select a colour from its computer colours' palette. The Dialog will have a SWT ColorDialog linked to it.
  - **Parameters**
    - `shell` - Shell, main window of the application at which the dialog will be attached.

    - `lProperties` - ArrayList<GladeProperty>, list of configuration values for the colorselection dialog.
  - **Returns** - SwtDialog, data of the new ColorSelection dialog.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog` (in 5.1.6, page 109)

- *createDialog*
  `public SwtDialog` **createDialog**`( Shell` **shell**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties** `)`
  - **Usage**
    - Creates a new custom Dialog of type DiALOG. The custom dialog will have a SWT Dialog linked to it.
  - **Parameters**
    - `shell` - Shell, main window of the application at which the dialog will be attached.
    - `name` - String, unique name of the Dialog.
    - `lProperties` - ArrayList<GladeProperty>, list of configuration values for the custom dialog.
  - **Returns** - SwtDialog, data of the new custom dialog.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog` (in 5.1.6, page 109)

- *createFileChooserDialog*
  `public SwtDialog` **createFileChooserDialog**`( Shell` **shell**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties** `)`
  - **Usage**
    - Creates a new Dialog of type FILECHOOSER. This dialog allows the user to select a file from its computer filesystem. The Dialog will have a SWT FileDialog linked to it.
  - **Parameters**
    - `shell` - Shell, main window of the application at which the dialog will be attached.
    - `name` - String, unique name of the Dialog.
    - `lProperties` - ArrayList<GladeProperty>, list of configuration values for the filechooser dialog.
  - **Returns** - SwtDialog, data of the new FileChooser dialog.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog` (in 5.1.6, page 109)

- *createFontSelectionDialog*
  `public SwtDialog` **createFontSelectionDialog**`( Shell` **shell**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties** `)`
  - **Usage**
    - Creates a new Dialog of type FONTSELECTION. This dialog allows the user to select a font from its computer available set of fonts. The Dialog will have a SWT FontDialog linked to it.
  - **Parameters**
    - `shell` - Shell, main window of the application at which the dialog will be attached.

    - `lProperties` - ArrayList<GladeProperty>, list of configuration values for the fontselection dialog.
  - **Returns** - SwtDialog, data of the new FontSelection dialog.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtDialog` (in 5.1.6, page 109)

- *createTopLevelWindow*
  `public SwtTopLevelWindow` **createTopLevelWindow**`( java.lang.String` **name**, `java.util.ArrayList` **properties**)
  - **Usage**
    - Creates a a new toplevel window. This new toplevel window will have a SWT shell linked to it.
  - **Parameters**
    - `name` - String, unique identifier of the toplevel window.
    - `properties` - ArrayList<GladeProperty>, list of configuration values for the toplevel window.
  - **Returns** - SwtTopLevelWindow, data of the new toplevel window.
  - **See Also**
    - `com.imagos.javaglade.toolkit.swt.SwtTopLevelWindow` (in 5.1.9, page 135)

- *drawingArea*
  `public Widget` **drawingArea**`( Composite` **parent**, `java.lang.String` **name**, `java.util.ArrayList` **lProperties**, `com.imagos.javaglade.vo.GladePacking` **packing**)
  - **Usage**
    - Build a widget of type DRAWING AREA (Canvas).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *emptyPanel*
  `public Widget` **emptyPanel**`( Composite` **parent**, `java.lang.String` **name**)
  - **Usage**
    - Build a widget of type COMPOSITE with no components inside.
  - **Parameters**
    - `parent` - Composite, parent of the empty composite widget.
    - `name` - String, name of the widget.
  - **Returns** - Widget, the widget built.

- *entry*
  `public Widget` **entry**`( Composite` **parent**, `java.lang.String` **name**, `java.util.ArrayList` **lProperties**, `com.imagos.javaglade.vo.GladePacking` **packing**)

    - Build a widget of type ENTRY (Text).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *entryComboBox*
  `public Widget` **entryComboBox**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type ENTRY COMBOBOX.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *fixedComposite*
  `public Composite` **fixedComposite**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type FIXED COMPOSITE (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *getBackgroundColor*
  `public int` **getBackgroundColor**`( )`
  - **Usage**
    - Gets the default widget background color.
  - **Returns** - int, default widget background color.

- *getDisplay*
  `public static Display` **getDisplay**`( )`
  - **Usage**
    - Gets the SWT Display instance used for the GUI construction.
  - **Returns** - Display, SWT display instance used for the GUI construction.

- *getImage*

  - **Usage**
    - Load the image from the file specified.
  - **Parameters**
    - `filename` - String, name of the image file.
  - **Returns** - Image, image loaded.

- *getResourceDir*
  `public String` **getResourceDir**`( )`
  - **Usage**
    - Gets the resources directory path.
  - **Returns** - String, resources directory path.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *group*
  `public Widget` **group**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type FRAME (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *hbox*
  `public Composite` **hbox**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type HORIZONTAL BOX (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *hbuttonbox*
  `public Composite` **hbuttonbox**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking`

- **Usage**
  - Build a widget of type HORIZONTAL BUTTON BOX (container). It's similar to a HORIZONTAL BOX widget but it also adds an own layout to simulate its own layout_display property.
- **Parameters**
  - `parent` - Composite, container widget.
  - `name` - String, name of the widget.
  - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
  - `packing` - GladePacking, all the packing properties for the widget.
- **Returns** - Widget, the widget built.

- *hpane*
  `public Composite` **hpane**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type HORIZONTAL PANE (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *hscale*
  `public Widget` **hscale**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type HORIZONTAL SCALE.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *hseparator*
  `public Widget` **hseparator**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type HORIZONTAL SEPARATOR.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.

- **Returns** - Widget, the widget built.

- *image*
  `public Widget` **image**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type IMAGE.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *label*
  `public Widget` **label**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type LABEL.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *menuBar*
  `public Widget` **menuBar**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type MENU BAR.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *menuItem*
  `public Widget` **menuItem**`( Widget` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**

- **Parameters**
  - `parent` - Widget, container widget.
  - `name` - String, name of the widget.
  - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
  - `packing` - GladePacking, all the packing properties for the widget.
- **Returns** - Widget, the widget built.

- *menuItemPopup*

  `public Widget` **menuItemPopup**`( Widget` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type MENU ITEM (submenu).
  - **Parameters**
    - `parent` - Widget, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *menuItemSeparator*

  `public Widget` **menuItemSeparator**`( Widget` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type MENU ITEM SEPARATOR.
  - **Parameters**
    - `parent` - Widget, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *popupMenu*

  `public Widget` **popupMenu**`( Widget` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type MENU BAR.
  - **Parameters**
    - `parent` - Widget, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *printSWTConstant*

- **Usage**
    - Prints out the text value of a SWT generic constant.
- **Parameters**
    - `value` - int, value of the SWT constant.
- **Returns** - String, description text of the SWT generic constant.

- *printWidgetInfo*
`public String` **printWidgetInfo**`( java.lang.String` **widgetname**`, Widget` **w**`, java.lang.String` **prefix** `)`
    - **Usage**
        - Formats a widget's layout information into a String, suitable for showing on screen.
    - **Parameters**
        - `widgetname` - String, unique name that identifies the widget in the context of the interface.
        - `w` - Widget, SWT widget to print out.
        - `prefix` - String, prefix to append to the resulting String. It's used for get a correct indentation.
    - **Returns** - String, widget's layout data formatted as a String.

- *printWidgetInfo*
`public String` **printWidgetInfo**`( Widget` **w**`, java.lang.String` **prefix** `)`
    - **Usage**
        - Formats a widget's layout information into a String, suitable for showing on screen.
    - **Parameters**
        - `w` - Widget, SWT widget to print out.
        - `prefix` - String, prefix to append to the resulting String. It's used for get a correct indentation.
    - **Returns** - String, widget's layout data formatted as a String.

- *progressBar*
`public Widget` **progressBar**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
    - **Usage**
        - Build a widget of type PROGRESS BAR.
    - **Parameters**
        - `parent` - Composite, container widget.
        - `name` - String, name of the widget.
        - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
        - `packing` - GladePacking, all the packing properties for the widget.
    - **Returns** - Widget, the widget built.

- *radioButton*
`public Widget` **radioButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

    - Build a widget of type RADIOBUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *radioGroup*
  `public Widget` **radioGroup**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type RADIO GROUP (container).
  - **Parameters**
    - `parent` - Composite, container widget;
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *scrolledComposite*
  `public Composite` **scrolledComposite**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type SCROLLED BOX (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *setBackgroundColor*
  `public void` **setBackgroundColor**`( int` **backgroundColor** `)`
  - **Usage**
    - Sets up the default widget background color.
  - **Parameters**
    - `backgroundColor` - int, default widget background color.

- *setButtonImage*
  `public void` **setButtonImage**`( Widget` **button**`, java.lang.String` **imagePath** `)`
  - **Usage**
    - Accesor method for setting up the icon of a SWT button.
  - **Parameters**

- ∗ `imagePath` - String, path of the image file of the icon.

- *setButtonLabel*
  `public void` **setButtonLabel**`( Widget` **button**`, java.lang.String` **label** `)`
  - **Usage**
    - ∗ Accesor method for setting up the label of a SWT button.
  - **Parameters**
    - ∗ `button` - Widget, SWT button widget.
    - ∗ `label` - String, new button's label text.

- *setMenuItemImage*
  `public void` **setMenuItemImage**`( Widget` **menuItem**`, java.lang.String` **imagePath** `)`
  - **Usage**
    - ∗ Sets up the icon of a SWT menuItem widget.
  - **Parameters**
    - ∗ `menuItem` - Widget, SWT menuItem widget.
    - ∗ `imagePath` - String, path of the menuItem's icon image file.

- *setMenuItemLabel*
  `public void` **setMenuItemLabel**`( Widget` **menuItem**`, java.lang.String` **label** `)`
  - **Usage**
    - ∗ Sets up the label of a SWT menuItem widget.
  - **Parameters**
    - ∗ `menuItem` - Widget, SWT menuItem widget.
    - ∗ `label` - String, new menuItem's label text.

- *setMenuItemPopupMenu*
  `public void` **setMenuItemPopupMenu**`( Widget` **menuItem**`, Widget` **popup** `)`
  - **Usage**
    - ∗ Sets up a popup menu linked to a SWT menuItem widget. In the SWT model for Menus, the way of making submenus is assigning a PopupMenu to a MenuItem. The popup menu widget must be of type Menu.
  - **Parameters**
    - ∗ `menuItem` - Widget, SWT menuItem widget.
    - ∗ `popup` - Widget, Popup menu widget that will be linked to the menuItem.
  - **See Also**
    - ∗ `org.eclipse.swt.widgets.Menu`

- *setResourceDir*
  `public void` **setResourceDir**`( java.lang.String` **resourceDir** `)`
  - **Usage**
    - ∗ Sets up the resources directory path.
  - **Parameters**

- *setTabItemLabel*
  `public void` **setTabItemLabel**`( Widget` **tabItem**`, java.lang.String` **label** `)`
  - **Usage**
    - Sets up the label of a SWT tabItem widget.
  - **Parameters**
    - `tabItem` - Widget, SWT tabItem widget.
    - `label` - String, new tabItem's label text.

- *spinButton*
  `public Widget` **spinButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type SPINBUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *statusBar*
  `public Widget` **statusBar**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type STATUS BAR.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *tabFolder*
  `public Widget` **tabFolder**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TAB FOLDER (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.

- *tabItem*
  public Widget **tabItem**(Widget **parent**, java.lang.String **name**)
  - **Usage**
    - Build an empty widget of type TAB ITEM.
  - **Parameters**
    - `parent` - Widget, container widget.
    - `name` - String, name of the widget.
  - **Returns** - Widget, the widget built.
- *tabItem*
  public Widget **tabItem**(Widget **parent**, java.lang.String **name**, java.util.ArrayList **lProperties**, com.imagos.javaglade.vo.GladePacking **packing**)
  - **Usage**
    - Build a widget of type TAB ITEM.
  - **Parameters**
    - `parent` - Widget, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.
- *table*
  public Composite **table**(Composite **parent**, java.lang.String **name**, java.util.ArrayList **properties**, com.imagos.javaglade.vo.GladePacking **packing**)
  - **Usage**
    - Build a widget of type TABLE BOX (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `properties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.
- *textArea*
  public Widget **textArea**(Composite **parent**, java.lang.String **name**, java.util.ArrayList **lProperties**, com.imagos.javaglade.vo.GladePacking **packing**, boolean **autoListener**)
  - **Usage**
    - Build a widget of type TEXT-AREA.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.

    - `autoListener` - boolean, True if an automatic Modify Listener should be added to the textArea component to adapt the widget to the contents; False in case the automatic Modify Listener mustn't be created (usually because another differente implementation of Modify listener is provided). <br/>The automatic listener is added when the TextArea widget is embedded in a Scrolled Composite holder. In this case the listener will automatically update the widget's size to fit the contents typed by the user.
  - **Returns** - Widget, the widget built.

- *toggleButton*
  `public Widget` **toggleButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOGGLEBUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolBar*
  `public Widget` **toolBar**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOOL BAR (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolButton*
  `public Widget` **toolButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOOL BUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolButton*
  `public Widget` **toolButton**`( Composite` **parent**`, java.lang.String` **name**`, java.lang.String` **label**`, java.lang.String` **imageName** `)`
  - **Usage**
    - Build a widget of type TOOL BUTTON with all the styles possible (SWT.CHECK — SWT.PUSH — SWT.RADIO — SWT.SEPARATOR — SWT.DROP_DOWN)
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `label` - String, text of the tool button.
    - `imageName` - String, image of the tool button.
  - **Returns** - Widget, the widget built.

- *toolItem*
  `public Widget` **toolItem**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOOL ITEM.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolRadioButton*
  `public Widget` **toolRadioButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOOL RADIO BUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolSeparator*
  `public Widget` **toolSeparator**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TOOL SEPARATOR.
  - **Parameters**

    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *toolToggleButton*

  `public Widget` **toolToggleButton**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type TOOL TOGGLE BUTTON.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *tree*

  `public Widget` **tree**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type TREE.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *treeItem*

  `public Widget` **treeItem**`( TreeItem` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

  - **Usage**
    - Build a widget of type TREE ITEM.
  - **Parameters**
    - `parent` - TreeItem, parent tree item widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *treeItem*

  `public Widget` **treeItem**`( TreeItem` **parent**`, java.lang.String` **name**`, java.lang.String [] ` **text**`, java.lang.String` **icon**`, int` **index** `)`

    - Build a widget of type TREE ITEM for a multicolumn tree.
  - **Parameters**
    - `parent` - TreeItem, parent tree item widget.
    - `name` - String, name of the widget.
    - `text` - String[], list of columns text for the tree item.
    - `icon` - String, image filename of the icon of the tree item.
    - `index` - int, position at which the tree item will be inserted. If index <0 then the item will be inserted at the end.
  - **Returns** - Widget, the widget built.

- *treeItem*

  `public Widget` **treeItem**`( TreeItem` **parent**`, java.lang.String` **name**`, java.lang.String` **text**`, java.lang.String` **icon**`, int` **index** `)`
  - **Usage**
    - Build a widget of type TREE ITEM.
  - **Parameters**
    - `parent` - TreeItem, parent tree item widget.
    - `name` - String, name of the widget.
    - `text` - String, text of the tree item.
    - `icon` - String, image filename of the icon of the tree item.
    - `index` - int, position at which the tree item will be inserted. If index <0 then the item will be inserted at the end.
  - **Returns** - Widget, the widget built.

- *treeItem*

  `public Widget` **treeItem**`( Tree` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type TREE ITEM.
  - **Parameters**
    - `parent` - Tree, container tree widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *treeItem*

  `public Widget` **treeItem**`( Tree` **parent**`, java.lang.String` **name**`, java.lang.String []` **text**`, java.lang.String` **icon**`, int` **index** `)`
  - **Usage**
    - Build a widget of type TREE ITEM for a multi-column tree.
  - **Parameters**
    - `parent` - Tree, container tree widget.
    - `name` - String, name of the widget.
    - `text` - String[], list of columns text for the tree item.
    - `icon` - String, image filename of the icon of the tree item.
    - `index` - int, position at which the tree item will be inserted. If index <0 then the

  - **Returns** - Widget, the widget built.

- *treeItem*
  `public Widget` **treeItem**`( Tree` **parent**`, java.lang.String` **name**`, java.lang.String` **text**`, java.lang.String` **icon**`, int` **index** `)`
  - **Usage**
    - Build a widget of type TREE ITEM.
  - **Parameters**
    - `parent` - Tree, container tree widget.
    - `name` - String, name of the widget.
    - `text` - String, text of the tree item.
    - `icon` - String, image filename of the icon of the tree item.
    - `index` - int, position at which the tree item will be inserted. If index <0 then the item will be inserted at the end.
  - **Returns** - Widget, the widget built.

- *vbox*
  `public Composite` **vbox**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type VERTICAL BOX (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *viewport*
  `public Composite` **viewport**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type VIEWPORT (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *vpane*
  `public Composite` **vpane**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`

    - Build a widget of type VERTICAL PANE (container).
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *vscale*
  `public Widget` **vscale**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type VERTICAL SCALE.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

- *vseparator*
  `public Widget` **vseparator**`( Composite` **parent**`, java.lang.String` **name**`, java.util.ArrayList` **lProperties**`, com.imagos.javaglade.vo.GladePacking` **packing** `)`
  - **Usage**
    - Build a widget of type VERTICAL SEPARATOR.
  - **Parameters**
    - `parent` - Composite, container widget.
    - `name` - String, name of the widget.
    - `lProperties` - ArrayList<GladeProperty>, list of properties of the widget.
    - `packing` - GladePacking, all the packing properties for the widget.
  - **Returns** - Widget, the widget built.

5.1.9 CLASS SwtTopLevelWindow

Implementation of the GUI widget TopLevelWindow for SWT. This widget is an abstract definition of a window in the SWT Toolkit. It makes use of the Shell widget implement by the SWT library. It enhances the Shell implementation made in SWT with methods for running and registering the events. <br/>Responsibilities:

- Provide a generic class for encapsulating any SWT widget that may act as a toplevel window, i.e. that hangs directly from the Glade interface and doesn't have a parent widget. This toplevel windows normally are showed in the starting of the application.</li>
- Provide methods for showing the window to the user and receive the event notifications.</li>

<br/>

DECLARATION

public class SwtTopLevelWindow
extends java.lang.Object
implements java.lang.Runnable

CONSTRUCTORS

- *SwtTopLevelWindow*
  `public` **SwtTopLevelWindow**`(` `java.lang.String` **name**, `java.util.ArrayList` **properties** `)`
  - **Usage**
    - Default SwtTopLevelWindow constructor.
  - **Parameters**
    - `name` - String, unique identifier of the TopLevel Window.
    - `properties` - ArrayList<GladeProperty>, list of toplevel window properties.

- *SwtTopLevelWindow*
  `public` **SwtTopLevelWindow**`(` `java.lang.String` **name**, `java.util.ArrayList` **properties**, `boolean` **isVisible** `)`
  - **Usage**
    - SwtTopLevelWindow constructor with visibility initialization.
  - **Parameters**
    - `name` - String, unique identifier of the TopLevel Window.
    - `properties` - ArrayList<GladeProperty>, list of toplevel window properties.
    - `isVisible` - boolean, Flag that indicates if the window must be rendered and showed to the user at the starting of the application.

- *SwtTopLevelWindow*
  `public` **SwtTopLevelWindow**`(` `java.lang.String` **name**, `boolean` **isVisible**, `int` **minHeight**, `int` **minWidth**, `java.lang.String` **image**, `java.lang.String` **title** `)`
  - **Usage**
    - SwtTopLevelWindow constructor with visibility initialization.
  - **Parameters**
    - `name` - String, unique identifier of the TopLevel Window.
    - `isVisible` - boolean, Flag that indicates if the window must be rendered and showed to the user at the starting of the application.
    - `minHeight` - int, minium height of the window in pixels.
    - `minWidth` - int, minium width of the window in pixels.
    - `image` - String, image icon for the window.
    - `title` - String, text for the title bar of the window.

- *SwtTopLevelWindow*
  `public` **SwtTopLevelWindow**`(` `java.lang.String` **name**, `Shell` **shell**, `boolean`

  - **Usage**
    - SwtTopLevelWindow constructor from an existent shell.
  - **Parameters**
    - `name` - String, unique identifier of the TopLevel Window.
    - `shell` - Shell, shell of the TopLevel Window.
    - `isVisible` - boolean,

METHODS

- *eventLoop*
  `public void` **eventLoop**`( )`
  - **Usage**
    - Makes the current toplevel window visible and inits the event reception mechanism.
- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Gets the name of the toplevel window.
  - **Returns** - String, unique identifier of the toplevel window.
- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.
- *getShell*
  `public Shell` **getShell**`( )`
  - **Usage**
    - Gets the rendered shell for the toplevel window.
  - **Returns** - Shell, rendered SWT shell for the window.
- *isVisible*
  `public boolean` **isVisible**`( )`
  - **Usage**
    - Gets the value of the isVisible flag that shows if the window is being showed to the user or not.
  - **Returns** - boolean, True if the window is showed to the user; False in any other case.
- *run*
  `public void` **run**`( )`
  - **Usage**
    - Starts and initializes the current toplevel window.
- *setName*

- **Usage**
  - Sets up the name of the toplevel window.
- **Parameters**
  - `name` - String, unique identifier of the toplevel window.

- *setVisible*
  `public void` **setVisible**`( boolean` **isVisible** `)`
  - **Usage**
    - Sets up the value of the isVisible flag that shows if the window is being showed to the user or not.
  - **Parameters**
    - `isVisible` - boolean, True if the window is showed to the user; False in any other case.

5.1.10 CLASS **SwtWidget**

DECLARATION

public class SwtWidget
extends java.lang.Object

FIELDS

- public static final int WIDGET_TYPE_WINDOW
  –
- public static final int WIDGET_TYPE_DIALOG
  –
- public static final int WIDGET_TYPE_WIDGET
  –

CONSTRUCTORS

- *SwtWidget*
  `public` **SwtWidget**`( java.lang.String` **widgetName**`, java.lang.Object` **widgetObj** `)`

METHODS

- *addChild*
  `public void` **addChild**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **child** `)`
- *delChild*
  `public void` **delChild**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **child** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getChildren*
  `public ArrayList` **getChildren**`( )`
- *getParent*
  `public SwtWidget` **getParent**`( )`
- *getWidget*
  `public Widget` **getWidget**`( )`
- *getWidgetName*
  `public String` **getWidgetName**`( )`
- *getWidgetObj*
  `public Object` **getWidgetObj**`( )`
- *setParent*
  `public void` **setParent**`( com.imagos.javaglade.toolkit.swt.SwtWidget` **parent** `)`
- *setWidgetName*
  `public void` **setWidgetName**`( java.lang.String` **widgetName** `)`
- *setWidgetObj*
  `public void` **setWidgetObj**`( java.lang.Object` **widgetObj** `)`

5.1.11 CLASS **TableComposite**

Implementation of the GUI widget TableComposite for SWT. This widget is not directly supported for the SWT toolkit, so a custom widget must be created. The custom table composite widget will render table of composite placeholders (like GridLayout) for attaching other components inside. The difference amongst SWT GridLayout and this custom TableComposite widget is that SWT.GridLayout doesn't support a specific number of rows known in advance and TableComposite will support it. <br/>Responsibilities:

- Build a GUI widget representing a table of composite placeholders.</li>
- Give accessors for attaching and getting the attached widgets for any cell.</li>

<br/>

DECLARATION

public class TableComposite
extends Composite

CONSTRUCTORS

- *TableComposite*
  `public` **TableComposite**`(` `Composite` **parent**`,` `int` **rows**`,` `int` **columns**`,` `boolean` **homogeneous**`,` `int` **horizontalSpacing**`,` `int` **verticalSpacing** `)`
  - **Usage**
    - Default TableComposite constructor.
  - **Parameters**
    - `parent` - Composite, widget's parent.
    - `rows` - int, table's number of rows.
    - `columns` - int, table's number of columns.
    - `homogeneous` - boolean, True if all the cells will be rendered with the same size; False if each cell will have its own bounds.
    - `horizontalSpacing` - int, number of pixels used for horizontal spacing among the cells (separation of the columns).
    - `verticalSpacing` - int, number of pixels used for vertical spacing among the cells (separation of the rows).

METHODS

- *get*
  `public Composite` **get**`(` `int` **row**`,` `int` **column** `)`
  - **Usage**
    - Get the composite widget that holds the contents of the cell.
  - **Parameters**
    - `row` - int, number of the row.
    - `column` - int, number of the column.
  - **Returns** - Composite, composite widget that holds the contents of the cell.

- *getColumns*
  `public int` **getColumns**`( )`
  - **Usage**
    - Get the number of columns of the table widget
  - **Returns** - int, number of columns of the table.

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *getRows*
  `public int` **getRows**`( )`
  - **Usage**
    - Get the number of rows of the table widget

- *getTable*
  `public Composite` **getTable**`( )`
  - **Usage**
    - Get the matrix of cells that forms the table
  - **Returns** - Composite[][], matrix with the cells of the table.

- *getTableCell*
  `public Composite` **getTableCell**`( com.imagos.javaglade.vo.GladePacking` **packing**`, GridData` **layoutData** `)`
  - **Usage**
    - Get the table cell corresponding to a child widget. In GTK the child widgets of a GtkTable are linked to the cells via the packing properties. In the packing properties a widget may be assigned to one of several of these cells according to the values setted for the parameters: top-attach, bottom-attach (number of rows where the child widget is placed) and left-attach, right-attach (number of columns where the child widget is placed). So the widget is placed in the area bounded by these values.

      It also updates the layout data of the cell, according to the layout data of the widget contained in the cell.
  - **Parameters**
    - `packing` - GladePacking, all the packing properties for the widget.
    - `layoutData` - GridData, layout data for the cell (it's the layout data specified for the content of the cell).
  - **Returns** - Composite, composite widget that holds the contents of the cell.

- *isHomogeneous*
  `public boolean` **isHomogeneous**`( )`
  - **Usage**
    - Gets the value of the homogeneous flag.
  - **Returns** - boolean, True if all the cells are homogeneous, i.e. if they are rendered with the same bounds; False in any other case.

- *toString*
  `public String` **toString**`( )`
  - **Usage**
    - Converts the table data into String format, suitable for being showed on the screen.
  - **Returns** - String, formatted String with the tablecomposite data.

METHODS INHERITED FROM CLASS `Composite`

Chapter 6

Package com.imagos.javaglade.util

Package Contents *Page*

Classes

6.1 Classes

6.1.1 CLASS **Indenter**

Utility class for indenting a paragraph's text. <br/>Responsibilities:

- Provide the application with an indentation mechanism, compatible with both blank characters or tab characters.</li>

<br/>

DECLARATION

public class Indenter
extends java.lang.Object

METHODS

- *getRevision*
  `public static String` **getRevision**`( )`
  - **Usage**
    - Returns tag Id assigned to CVS source file.

- *indent*
  `public static String` **indent**`( int` **level**`, java.lang.String` **str** `)`
  - **Usage**
    - Utility method to indent a paragraph's text.
  - **Parameters**
    - `level` - int, number of indenting characters placed before each line of the text.
    - `str` - String, paragraph's text without indentation.
  - **Returns** - String, indented paragraph's text.

- *setUseTabs*
  `public static void` **setUseTabs**`( boolean` **useTabs** `)`
  - **Usage**
    - Sets up the useTabs flag. This method enables/disables de TABS mode.
  - **Parameters**
    - `useTabs` - boolean, True to enable the TABS mode, False to disable it.