| | **Class** | [**Tree**](http://docs.google.com/overview-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/BreezyGUI/Format.html)   [**NEXT CLASS**](http://docs.google.com/BreezyGUI/GBDialog.html) | [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/GBApplet.html) |
| SUMMARY:  INNER | [FIELD](#1fob9te) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL:  FIELD | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

## **BreezyGUI**

Class GBApplet

java.lang.Object  
 |  
 +--java.awt.Component  
 |  
 +--java.awt.Container  
 |  
 +--java.awt.Window  
 |  
 +--java.awt.Frame  
 |  
 +--**BreezyGUI.GBApplet**

public class **GBApplet**extends java.awt.Frame

The class GBApplet (short for Grid Bag Applet) provides a high-level applet window. The window comes with a built in GridBagLayout. Components are added to the window by specifying their row position, column position, width (in columns), and height (in rows)in the grid.

Events of different types, such as button clicks and list item selections, are handled by implementing the appropriate GBApplet methods in the application.

Unlike GBFrame, GBApplet does not support pull down menus.

Example use of GBApplet:

// Hello world applet with BreezyGUI.

import java.awt.\*;  
 import BreezyGUI.\*;

public class HelloWorld extends GBApplet{

// Add a labeled button to the window at row 1, column 1, width 1, height 1.  
 Button clickMe = addButton("Click me", 1, 1, 1, 1);

// Implement the method to handle the button click.  
 public void buttonClicked(Button buttonObj){  
 messageBox("Hello world!");  
 }  
 }

**See Also:** [Serialized Form](http://docs.google.com/serialized-form.html#BreezyGUI.GBApplet)

| **Fields inherited from class java.awt.Frame** |
| --- |
| CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, HAND\_CURSOR, ICONIFIED, MOVE\_CURSOR, N\_RESIZE\_CURSOR, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, W\_RESIZE\_CURSOR, WAIT\_CURSOR |

| **Fields inherited from class java.awt.Component** |
| --- |
| BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT |

| **Constructor Summary** | |
| --- | --- |
| [**GBApplet**](http://docs.google.com/BreezyGUI/GBApplet.html#GBApplet())()            Creates a GBApplet. |

| **Method Summary** | |
| --- | --- |
| java.awt.Button | [**addButton**](http://docs.google.com/BreezyGUI/GBApplet.html#addButton(java.lang.String,%20int,%20int,%20int,%20int))(java.lang.String text, int row, int col, int width, int height)            Adds a Button with the specified name to the specified position, with the specified width and height. |
| java.awt.Checkbox | [**addCheckbox**](http://docs.google.com/BreezyGUI/GBApplet.html#addCheckbox(java.lang.String,%20int,%20int,%20int,%20int))(java.lang.String text, int row, int col, int width, int height)            Adds a CheckBox to the specified position, with the specified width and height. |
| java.awt.Choice | [**addChoice**](http://docs.google.com/BreezyGUI/GBApplet.html#addChoice(int,%20int,%20int,%20int))(int row, int col, int width, int height)            Adds a Choice (pull down list) to the specified position, with the specified width and height. |
| [DoubleField](http://docs.google.com/BreezyGUI/DoubleField.html) | [**addDoubleField**](http://docs.google.com/BreezyGUI/GBApplet.html#addDoubleField(double,%20int,%20int,%20int,%20int))(double num, int row, int col, int width, int height)            Adds a DoubleField containing the specified number to the specified position, with the specified width and height. |
| [IntegerField](http://docs.google.com/BreezyGUI/IntegerField.html) | [**addIntegerField**](http://docs.google.com/BreezyGUI/GBApplet.html#addIntegerField(int,%20int,%20int,%20int,%20int))(int num, int row, int col, int width, int height)            Adds an IntegerField containing the specified integer to the specified position, with the specified width and height. |
| java.awt.Label | [**addLabel**](http://docs.google.com/BreezyGUI/GBApplet.html#addLabel(java.lang.String,%20int,%20int,%20int,%20int))(java.lang.String text, int row, int col, int width, int height)            Adds a label with the specified name to the specified position, with the specified width and height. |
| java.awt.List | [**addList**](http://docs.google.com/BreezyGUI/GBApplet.html#addList(int,%20int,%20int,%20int))(int row, int col, int width, int height)            Adds a List (a scrolling list) to the specified position, with the specified width and height. |
| java.awt.TextArea | [**addTextArea**](http://docs.google.com/BreezyGUI/GBApplet.html#addTextArea(java.lang.String,%20int,%20int,%20int,%20int))(java.lang.String text, int row, int col, int width, int height)            Adds a TextArea containing the specified string to the specified position, with the specified width and height. |
| java.awt.TextField | [**addTextField**](http://docs.google.com/BreezyGUI/GBApplet.html#addTextField(java.lang.String,%20int,%20int,%20int,%20int))(java.lang.String text, int row, int col, int width, int height)            Adds a TextField containing the specified string to the specified position, with the specified width and height. |
| void | [**buttonClicked**](http://docs.google.com/BreezyGUI/GBApplet.html#buttonClicked(java.awt.Button))(java.awt.Button buttonObj)            The application must implement this method in order to handle button clicks. |
| void | [**listDoubleClicked**](http://docs.google.com/BreezyGUI/GBApplet.html#listDoubleClicked(java.awt.List,%20java.lang.String))(java.awt.List listObj, java.lang.String itemClicked)            The application must implement this method in order to handle a double click on an item in a scrolling list. |
| void | [**listItemSelected**](http://docs.google.com/BreezyGUI/GBApplet.html#listItemSelected(java.awt.List))(java.awt.List listObj)            The application must implement this method in order to handle a selection (single click) on an item in a scrolling list. |
| void | [**messageBox**](http://docs.google.com/BreezyGUI/GBApplet.html#messageBox(double))(double num)            Pops up a message box containing the number and an OK button. |
| void | [**messageBox**](http://docs.google.com/BreezyGUI/GBApplet.html#messageBox(java.lang.Object))(java.lang.Object obj)            Pops up a message box containing the string representation of the object and an OK button. |
| void | [**messageBox**](http://docs.google.com/BreezyGUI/GBApplet.html#messageBox(java.lang.String))(java.lang.String msg)            Pops up a message box containing the string and an OK button. |
| void | [**mouseClicked**](http://docs.google.com/BreezyGUI/GBApplet.html#mouseClicked(int,%20int))(int x, int y)            The application must implement this method in order to handle mouse clicks in the window. |
| void | [**mouseDragged**](http://docs.google.com/BreezyGUI/GBApplet.html#mouseDragged(int,%20int))(int x, int y)            The application must implement this method in order to handle mouse dragged events in the window. |
| void | [**mouseMoved**](http://docs.google.com/BreezyGUI/GBApplet.html#mouseMoved(int,%20int))(int x, int y)            The application must implement this method in order to handle mouse moved events in the window. |
| void | [**mousePressed**](http://docs.google.com/BreezyGUI/GBApplet.html#mousePressed(int,%20int))(int x, int y)            The application must implement this method in order to handle mouse pressed events in the window. |
| void | [**mouseReleased**](http://docs.google.com/BreezyGUI/GBApplet.html#mouseReleased(int,%20int))(int x, int y)            The application must implement this method in order to handle mouse released events in the window. |

| **Methods inherited from class java.awt.Frame** |
| --- |
| addNotify, finalize, getCursorType, getFrames, getIconImage, getMenuBar, getState, getTitle, isResizable, paramString, remove, removeNotify, setCursor, setIconImage, setMenuBar, setResizable, setState, setTitle |

| **Methods inherited from class java.awt.Window** |
| --- |
| addWindowListener, applyResourceBundle, applyResourceBundle, dispose, getFocusOwner, getInputContext, getLocale, getOwnedWindows, getOwner, getToolkit, getWarningString, hide, isShowing, pack, postEvent, processEvent, processWindowEvent, removeWindowListener, setCursor, show, toBack, toFront |

| **Methods inherited from class java.awt.Container** |
| --- |
| add, add, add, add, add, addContainerListener, addImpl, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getInsets, getLayout, getMaximumSize, getMinimumSize, getPreferredSize, insets, invalidate, isAncestorOf, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, remove, removeAll, removeContainerListener, setFont, setLayout, update, validate, validateTree |

| **Methods inherited from class java.awt.Component** |
| --- |
| action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputMethodRequests, getLocation, getLocation, getLocationOnScreen, getName, getParent, getPeer, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setDropTarget, setEnabled, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, size, toString, transferFocus |

| **Methods inherited from class java.lang.Object** |
| --- |
| clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait |

| **Constructor Detail** |
| --- |

### GBApplet

public **GBApplet**()

Creates a GBApplet.

| **Method Detail** |
| --- |

### messageBox

public void **messageBox**(double num)

Pops up a message box containing the number and an OK button.

### messageBox

public void **messageBox**(java.lang.String msg)

Pops up a message box containing the string and an OK button.

### messageBox

public void **messageBox**(java.lang.Object obj)

Pops up a message box containing the string representation of the object and an OK button.

### listItemSelected

public void **listItemSelected**(java.awt.List listObj)

The application must implement this method in order to handle a selection (single click) on an item in a scrolling list. If no event handling is desired, this method need not be implemented. **Parameters:**listObj - The List in which the selection occurred.

### buttonClicked

public void **buttonClicked**(java.awt.Button buttonObj)

The application must implement this method in order to handle button clicks. If no buttons are added to the window, this method need not be implemented. **Parameters:**buttonObj - The Button in which the click occurred. Example:

// Add the buttons to the window.  
 Button okButton = addButton("OK", 1, 1, 1, 1);  
 Button cancelButton = addButton("Cancel", 1, 2, 1, 1);  
 // Handle the button selection.  
 public void buttonClicked(Button buttonObj){  
 if (buttonObj == okButton)  
 messageBox("OK selected");  
 else  
 messageBox("Cancel selected");  
 }

### listDoubleClicked

public void **listDoubleClicked**(java.awt.List listObj,  
 java.lang.String itemClicked)

The application must implement this method in order to handle a double click on an item in a scrolling list. If no double click is anticipated, this method need not be implemented. **Parameters:**listObj - The List in which the double click occurred.itemClicked - The string selected by the double click.

### mouseClicked

public void **mouseClicked**(int x,  
 int y)

The application must implement this method in order to handle mouse clicks in the window. If no event handling is desired, this method need no be implemented. **Parameters:**x - The x coordinate of the mouse in the window.y - The y coordinate of the mouse in the window.

### mousePressed

public void **mousePressed**(int x,  
 int y)

The application must implement this method in order to handle mouse pressed events in the window. If no event handling is desired, this method need no be implemented. **Parameters:**x - The x coordinate of the mouse in the window.y - The y coordinate of the mouse in the window.

### mouseReleased

public void **mouseReleased**(int x,  
 int y)

The application must implement this method in order to handle mouse released events in the window. If no event handling is desired, this method need no be implemented. **Parameters:**x - The x coordinate of the mouse in the window.y - The y coordinate of the mouse in the window.

### mouseMoved

public void **mouseMoved**(int x,  
 int y)

The application must implement this method in order to handle mouse moved events in the window. If no event handling is desired, this method need no be implemented. **Parameters:**x - The x coordinate of the mouse in the window.y - The y coordinate of the mouse in the window.

### mouseDragged

public void **mouseDragged**(int x,  
 int y)

The application must implement this method in order to handle mouse dragged events in the window. If no event handling is desired, this method need no be implemented. **Parameters:**x - The x coordinate of the mouse in the window.y - The y coordinate of the mouse in the window.

### addLabel

public java.awt.Label **addLabel**(java.lang.String text,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a label with the specified name to the specified position, with the specified width and height. **Parameters:**text - The name of the label.row - The beginning row (starting from 1) of the window's grid in which the label is displayed.col - The beginning column (starting from 1) of the window's grid in which the label is displayed.width - The number of columns of the window's grid occuppied by the label.height - The number of rows of the window's grid occuppied by the label.**Returns:**the Label

### addButton

public java.awt.Button **addButton**(java.lang.String text,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a Button with the specified name to the specified position, with the specified width and height. **Parameters:**text - The name of the Button.row - The beginning row (starting from 1) of the window's grid in which the Button is displayed.col - The beginning column (starting from 1) of the window's grid in which the Button is displayed.width - The number of columns of the window's grid occuppied by the Button.height - The number of rows of the window's grid occuppied by the Button.**Returns:**the button. Example:

Button okButton = addButton("OK", 1, 1, 1, 1);  
 // Adds a button at position row 1, column 1,  
 // with a width of 1 column and a height of one row.

### addTextField

public java.awt.TextField **addTextField**(java.lang.String text,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a TextField containing the specified string to the specified position, with the specified width and height. **Parameters:**text - The string to be displayed initially.row - The beginning row (starting from 1) of the window's grid in which the TextField is displayed.col - The beginning column (starting from 1) of the window's grid in which the TextField is displayed.width - The number of columns of the window's grid occuppied by the TextField.height - The number of rows of the window's grid occuppied by the TextField.**Returns:**the TextField.

### addTextArea

public java.awt.TextArea **addTextArea**(java.lang.String text,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a TextArea containing the specified string to the specified position, with the specified width and height. **Parameters:**text - The string to be displayed initially.row - The beginning row (starting from 1) of the window's grid in which the TextArea is displayed.col - The beginning column (starting from 1) of the window's grid in which the TextArea is displayed.width - The number of columns of the window's grid occuppied by the TextArea.height - The number of rows of the window's grid occuppied by the TextArea.**Returns:**the TextArea.

### addList

public java.awt.List **addList**(int row,  
 int col,  
 int width,  
 int height)

Adds a List (a scrolling list) to the specified position, with the specified width and height. **Parameters:**row - The beginning row (starting from 1) of the window's grid in which the List is displayed.col - The beginning column (starting from 1) of the window's grid in which the List is displayed.width - The number of columns of the window's grid occuppied by the List.height - The number of rows of the window's grid occuppied by the List.**Returns:**the List

### addChoice

public java.awt.Choice **addChoice**(int row,  
 int col,  
 int width,  
 int height)

Adds a Choice (pull down list) to the specified position, with the specified width and height. **Parameters:**row - The beginning row (starting from 1) of the window's grid in which the Choice is displayed.col - The beginning column (starting from 1) of the window's grid in which the Choice is displayed.width - The number of columns of the window's grid occuppied by the Choice.height - The number of rows of the window's grid occuppied by the Choice.**Returns:**the Choice.

### addIntegerField

public [IntegerField](http://docs.google.com/BreezyGUI/IntegerField.html) **addIntegerField**(int num,  
 int row,  
 int col,  
 int width,  
 int height)

Adds an IntegerField containing the specified integer to the specified position, with the specified width and height. **Parameters:**num - The integer to be displayed initially.row - The beginning row (starting from 1) of the window's grid in which the IntegerField is displayed.col - The beginning column (starting from 1) of the window's grid in which the IntegerField is displayed.width - The number of columns of the window's grid occuppied by the IntegerField.height - The number of rows of the window's grid occuppied by the IntegerField.

### addDoubleField

public [DoubleField](http://docs.google.com/BreezyGUI/DoubleField.html) **addDoubleField**(double num,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a DoubleField containing the specified number to the specified position, with the specified width and height. **Parameters:**num - The number to be displayed initially.row - The beginning row (starting from 1) of the window's grid in which the DoubleField is displayed.col - The beginning column (starting from 1) of the window's grid in which the DoubleField is displayed.width - The number of columns of the window's grid occuppied by the DoubleField.height - The number of rows of the window's grid occuppied by the DoubleField.

### addCheckbox

public java.awt.Checkbox **addCheckbox**(java.lang.String text,  
 int row,  
 int col,  
 int width,  
 int height)

Adds a CheckBox to the specified position, with the specified width and height. **Parameters:**row - The beginning row (starting from 1) of the window's grid in which the CheckBox is displayed.col - The beginning column (starting from 1) of the window's grid in which the CheckBox is displayed.width - The number of columns of the window's grid occuppied by the CheckBox.height - The number of rows of the window's grid occuppied by the CheckBox.**Returns:**the CheckBox.

| | **Class** | [**Tree**](http://docs.google.com/overview-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/BreezyGUI/Format.html)   [**NEXT CLASS**](http://docs.google.com/BreezyGUI/GBDialog.html) | [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/GBApplet.html) |
| SUMMARY:  INNER | [FIELD](#1fob9te) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL:  FIELD | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |