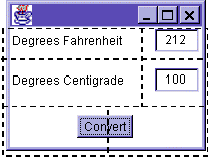
# **Laying Out Window Controls**

Window controls are laid out by using an imaginary grid superimposed on the window.  For instance, consider the interface for the simplified temperature conversion program shown next and consider the grid that has been superimposed on the window.



The grid has six cells arranged in three rows and two columns.  BreezyGUI automatically adjusts a cell's width to the size of the control occupying the cell. Although most controls occupy a single cell, a control can span several cells.  A cell’s location in the grid is given as an ordered pair – (row number, column number) – thus, the label “Degrees Centigrade” is in cell (2,1) and the number “100” is in cell (2,2). The **Convert** button spans two cells, namely cells (3,1) and (3,2).

The syntax for defining a window control is:

<class of control> <variable name of control> =

<add method> (<initial value>,

<row #>, <column #>,

<width in cells>, <height in cells>);

Although controls are located with respect to a grid, we never specify the dimensions of this grid explicitly. BreezyGUI infers the dimensions of the grid from the location and extent information provided. Thus, to lay out a window, simply indicate the position and extent of the controls, and the grid will automatically adjusts itself to the needed number of rows and columns.

Here is the code needed to declare the controls in the temperature conversions program above. An add method instantiate a control and specifies each control's initial value and position in the grid:

Label degreesFahrenheitLabel = addLabel ("Degrees Fahrenheit",1,1,1,1);

Label degreesCentigradeLabel = addLabel ("Degrees Centigrade",2,1,1,1);

IntegerField degreesFahrenheitField = addIntegerField (0,1,2,1,1);

IntegerField degreesCentigradeField = addIntegerField (0,2,2,1,1);

Button convertButton = addButton ("Convert",3,1,2,1);

These lines of code adhere to the following format:

Label <name> = addLabel ("..." ,r,c,w,h);

IntegerField <name> = addIntegerField (<integer>,r,c,w,h);

Button <name> = addButton ("..." ,r,c,w,h);

Each control has the following attributes:

* **Class or type:** There are several types, some of which we see here and some of which will be introduced later. The types used in the temperature conversion program are
  + **Label:** Used to prompt the user and explain the window's purpose.
  + **IntegerField:** Used to enter or display integer values. If the user enters nonnumeric or floating point data in an integer field, the program automatically converts it to 0 before use.
  + **Button:** Used to signal a command to the program.
* **Variable name** by which the control is known in the program.
* **Initial value** of the control.
* **Location in the grid:** The location is specified in terms of the cell that contains the top left corner of the control. Thus, the location of the Convert button is (3,1).
* **Extent:** The number of horizontal and vertical cells occupied by the control, expressed as (width in cells, height in cells). Thus, the extent of all the objects, except the button, is (1,1). The button's extent is (2,1).

The next table shows the values of these attributes for each of the controls shown above:

| **Type of Window Control** | **Variable Name for Control in Program** | **Initial Value** | **Location** | **Extent** |
| --- | --- | --- | --- | --- |
| Label | degreesFahrenheitLabel | "Degrees Fahrenheit" | (1,1) | (1,1) |
| Label | degreesCentigradeLabel | "Degrees Centigrade" | (2,1) | (1,1) |
| IntegerField | degreesFahrenheitField | 0 | (1,2) | (1,1) |
| IntegerField | degreesCentigradeField | 0 | (2,2) | (1,1) |
| Button | convertButton | "Convert" | (3,1) | (2,1) |