

# APPLICATION OF GRIDBAGLAYOUT FOR JAVA GUI DEVELOPMENT IN ACCELERATOR CONTROL SYSTEM

Jianjun Hu  
Stefania Canella

IMP-CAS, Lanzhou, China, 730000  
INFN-LNL Legnaro(PD), Italy, 35020

## Abstract:

GridBagLayout is one of the most sophisticated, flexible layout manager the Java platform provides. It could lead to better overall GUI development and a more robust final product. In this article, three important topics about GridBagLayout will be mainly discussed on how to choose an appropriate layout manager, how to apply the GridBagLayout freely, and how to solve some common problems while using the GridBagLayout. At the end of this article, an example GUI of steerer control for ECR ion source with its layout structure tree is presented for showing how to organize the GUI primarily with GridBagLayout layout manager.

*Keywords:* Java, GUI, GridBagLayout, accelerator

supplies five commonly used layout managers: BorderLayout, BoxLayout, FlowLayout, GridLayout, and GridBagLayout. And the GridBagLayout is a very flexible one. These layout managers are designed for displaying multiple components at once, and are shown in the preceding figure. By default, each container has a layout manager—an object that performs layout management for the components within the container. Components can provide size and alignment hints to layout managers, but layout managers have the final say on the size and position of those components.

## 2 GENERAL RULES FOR USING LAYOUT MANAGER

### 1 INTRODUCTION

Layout management is the process of determining the size and position of components. The Java platform

Layout managers have different strengths and weaknesses. How to Choose a Layout Manager? The following table lists some common layout scenarios and which layout managers might work for each scenario.

<i>Scenario</i>	<i>Suggestion</i>
You need to display a component in as much space as it can get.	Consider using BorderLayout or GridBagLayout. If you use BorderLayout, you'll need to put the space-hungry component in the center. With GridBagLayout, you'll need to set the constraints for the component so that <code>fill=GridBagConstraints.BOTH</code> . Another possibility is to use BoxLayout, making the space-hungry component specify very large preferred and maximum sizes.
You need to display a few components in a compact row at their natural size.	Consider using a JPanel to group the components and using either the JPanel's default FlowLayout manager or the BoxLayout manager.
You need to display a few components of the same size in rows and columns.	GridLayout is perfect for this.
You need to display a few components in a row or column, possibly with varying amounts of space between them, custom alignment, or custom component sizes.	BoxLayout is perfect for this.
You have a complex layout with many components.	Consider either using GridBagLayout or grouping the components into one or more JPanels to simplify layout. Each JPanel might use a different layout manager.

If none of the layout managers we discuss is right for your situation, feel free to use other layout managers that you write or find. There are lots of open resource on

Layout Manager in WWW which are fit for your programming and waiting for you to mining.

### 3 APPLY GRIDBAGLAYOUT TO JAVA GUI DEVELOPMENT

GridBagLayout is the most flexible -- and complex -- layout manager the Java platform provides. A GridBagLayout places components in a grid of rows and columns, allowing specified components to span multiple rows or columns. Not all rows necessarily have the same height. Similarly, not all columns necessarily have the same width. Essentially, GridBagLayout places components in rectangles (cells) in a grid, and then uses the components' preferred sizes to determine how big the cells should be.

This figure 1 is a picture of the magnet field control Java GUI for ECR Ion Source in INFN-LNL. GridBagLayout has been used in every section of the GUI.

The way the program specifies the size and position characteristics of its components is by specifying

constraints for each component, To specify constraints, you set instance variables in a GridBagConstraints object and tell the GridBagLayout (with the setConstraints method) to associate the constraints with the component. You must be careful while you reset the GridBagConstraints instance variables. You can set the following GridBagConstraints instance variables: gridx, gridy ; gridwidth, gridheight ; fill; ipadx, ipady; insets; anchor; weightx, weighty. It is easy to find the explanation of the usage of these parameters in <http://java.sun.com>.

Although the GridBagLayout is very useful, we can use it together with other layout manager to make our programming flexible. The figure 2 is the GUI screen shoot of Java GUI for steerer control of ECR Ion Source in INFN-LNL. The FlowLayout, BorderLayout, BorderLayout and GridBagLayout have been used. Certainly, the GridBadLayout is the most frequently used layout manager.

### 4 CONCLUSION

In brief, using GridBagLayout layout manager can

lead to GUI that more closely meet the original specifications, while also closely meeting the expected run-time requirements for accelerator control system.

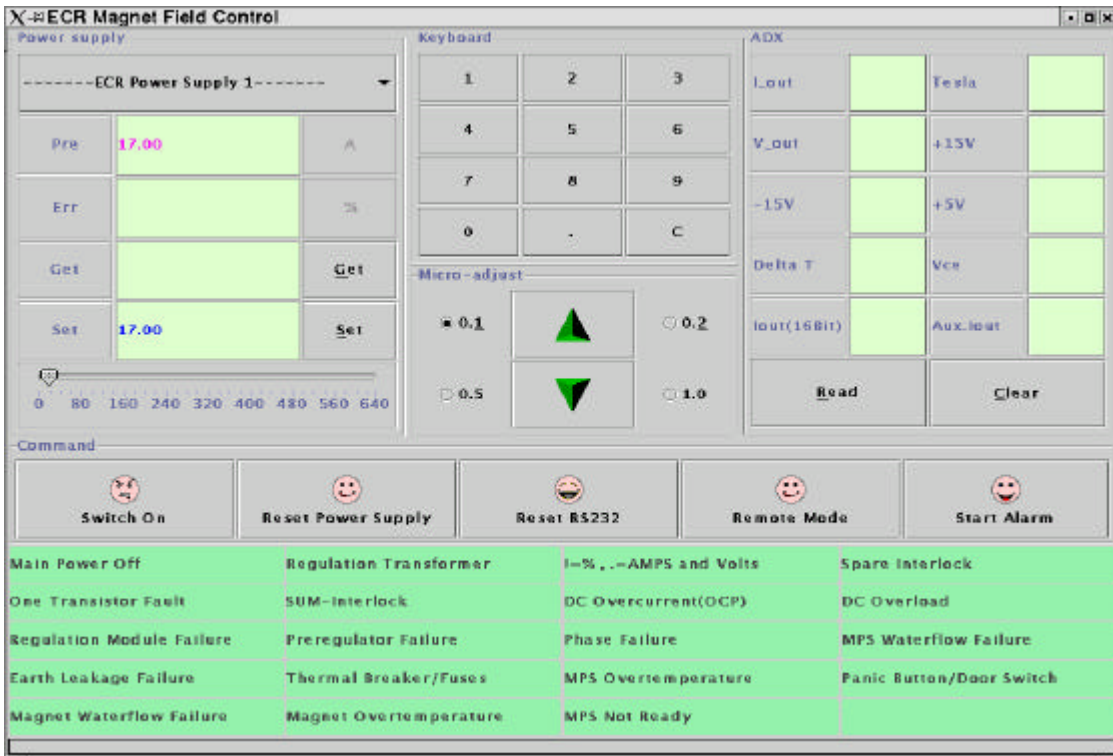


Figure1 Magnet Field Control GUI for ECR Ion Source

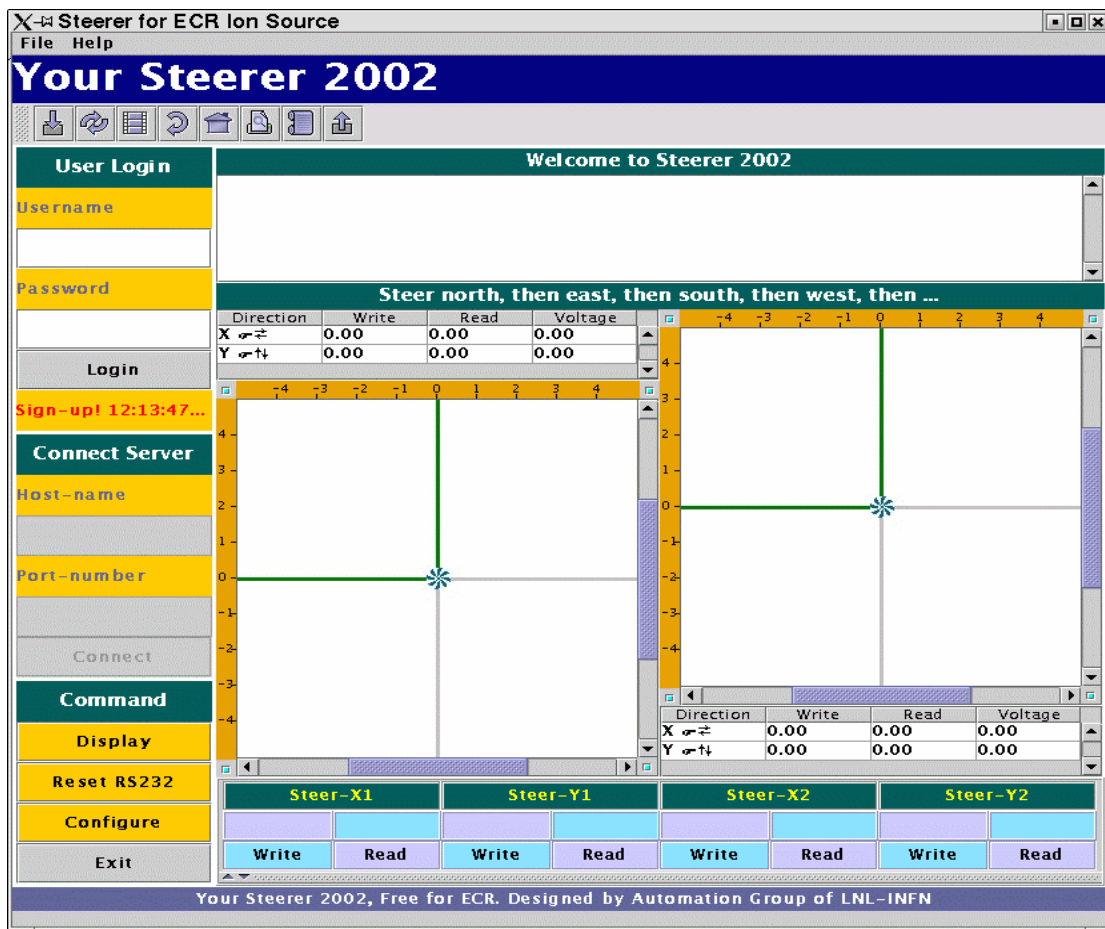


Figure 2 Steerer Control GUI for ECR Ion Source

## 5 ACKNOWLEDGEMENTS

The authors would like to thank Dr. Giorgio Bassato, Dr. Andrea Battistella, Dr. Davide Carlucci and Dr. Alessandro Zanon for their enthusiasm assistance and supports in many aspects.

## 6 REFERENCES

- [1] <http://java.sun.com>
- [2] G.Bassato, A. Battistella, S. Canella, Experience with Linux in the L.N.L. Accelerator Control Systems, INFN-LNL Annual Report 2001, 257