```
                                                //right is empty at first
        choicePanel = new JPanel();
        choicePanel.setBackground(Color.lightGray);
        p.add(choicePanel);

        setBounds(100, 100, 300, 200);
        setVisible(true);
}
```

In this simple program, we keep our three lists of investments in three Vectors—Stocks, Bonds, and Mutuals. We load them with arbitrary values as part of program initializing.

```
        Vector Mutuals = new Vector();
        Mutuals.addElement("Fidelity Magellan");
        Mutuals.addElement("T Rowe Price");
        Mutuals.addElement("Vanguard PrimeCap");
        Mutuals.addElement("Lindner Fund");
```

In a real system, we'd probably read them in from a file or database. Then, when the user clicks on one of the three investment types in the left list box, we pass the equivalent Vector to our Factory, which returns one of the Builders:

```
private void stockList_Click() {
    Vector v = null;
    int index = stockList.getSelectedIndex();
    choicePanel.removeAll(); //remove previous gui panel

    //switches between three different Vectors
    //and passes the selected one to the Builder pattern
    switch (index) {
    case 0:
        v = Stocks; break;
    case 1:
        v = Bonds; break;
    case 2:
        v = Mutuals;
    }
    mchoice = cfact.getChoiceUI(v);     //get one of the guis
    choicePanel.add(mchoice.getUI());   //insert on right
    choicePanel.validate();             //re-layout and display
    choicePanel.repaint ();
    Plot.setEnabled(true);              //allow plots
}
```

We show this *switch* code for simplicity. We could just as easily have attached a different ActionListener to each of the three buttons. We do save the multiChoice panel that the factory creates in the mchoice variable so that we can pass it to the Plot dialog.