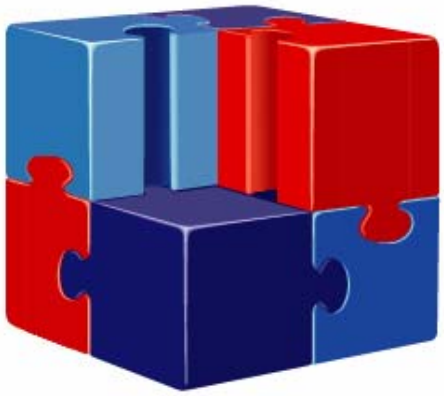
jProductivity LLC

Protect your investments with Components!



Components!

Calculator

User Guide

http://www.jproductivity.com

Contents

Highlights

Components! Calculator is a Java Component which can be integrated in a Java IDE and used in a GUI Java Application or Applet. Calculator component is designed to perform any of the standard operations for which user would normally use a handheld calculator.

Components! Calculator is a robust and sophisticated component for JFC/Swing:

- Essential for creating modern powerful and polished Java GUI application.
- Designed for speed and flexibility with advanced features and powerful performance
- Provide comprehensive presentation layer allowing developer to focus on the implementation
- Designed with Developer Needs in mind.
- Over the last several years millions of developers have embraced Java as their primary or secondary language. Wealth of available Java IDEs, utilization of Java in the development of Server-based, Desktop, Web and Mobile applications is a clear indication that Java is hear to stay. Java's strong support for the component-based development created a growing market of developers that come to rely on the component-based development approach. Utilization of JavaBeans components introduced rapid development of feature reach GUI interfaces and development cost efficiency.
- Built-in convenience.
- All jProductivity components design with the concept allowing developers to simply drop components on the form and start using it. However if customization is required, all jProductivity components are highly customizable and extendible.
- Customized display options.
- All aspects of a visual component could be highly customized. From simple Background and Foreground Color manipulation to complete replacement of the Components internal rendering engine with the custom one.
- On the fly validation.
- No mater what data type component is designed to work with, all jProductivity components validate its data on the fly and therefore protecting developer and end-user from accidental data corruption.
- Wide range of applications.
- Because all jProductivity Components are true 100% Java they could be used in any visual application environment – from web to desktop application.

High Level Description

The following topics briefly outline major Components! Calculator features and abilities.

Calculator package includes two components: Calculator itself and Calculator Field as described below.

Calculator

Calculator component is designed to perform any of the standard operations for which user would normally use a handheld calculator. Calculator performs basic arithmetic,

such as addition and subtraction, calculates reciprocal of the displayed number “1/x”, calculates square root of the displayed number, and performs percentage calculations.
However, Calculator could easily be extended by the end-developer to perform any types of unary or binary calculations. On addition to calculation functionality, Calculator component is able to store the displayed number in memory and allows several standard operations with stored number:

- Recall stored number
- Clear Memory
- Add displayed number to the number already in memory
- Subtract displayed number from the number already in memory

When user stores a number in the Calculator’s memory, an “**M”** symbol appears in the box above the memory options. If user stores another number, it replaces the one currently in memory.
Calculator component can present calculation result with different levels of precision – up to 11 decimal points. Number of fraction digits in the result field could be controlled by Fraction Digits Spinner, however all internal calculations are always performed with accuracy of 11 significant decimal digits!

Calculator Field

Calculator Field represents a field to show, enter and validate the number. The Calculator Field consists of the text field to show/enter the number and a button to popup the Calculator. Calculator can be also presented by pressing the Down Arrow key on the keyboard (can be redefined). Using popup calculator allows entering the number as result of some calculations. The text, tool tip and icon for the button can be changed as well.

Extending Calculator

While Calculator component provides ready to use functionality that is powerful enough in most cases, it’s also possible to extend Calculator component by providing additional operations. The following example illustrates implementation of a new operation and its addition to the Calculator component. We will create new “sine” operation that will allow us to calculate the sine of the displayed number.

Create a class that will handle “sine” operation

```
public class SinCalculatorOperation
    extends UnaryOperation
{
  public BigDecimal performOperation(Calculator aCalculator,
                                     BigDecimal aFirstOperand,
                                     BigDecimal aSecondOperand)
         Throws CalculatorException
  {
    BigDecimal result = null;

    try
    {
      result = new BigDecimal(Math.sin(Math.
                              toRadians(aSecondOperand.doubleValue())));
    }
    catch (Exception ex)
```

```
    {
      throw new CalculatorException("Error");
    }

    return result;
  }
}
```

Subclass the CalculatorPanel components

```
public class ExtCalculatorPanel extends CalculatorPanel
{
}
```

Create a button that will call the "sine" operation and add it to the ExtCalculatorPanel component.

```
JButton btnSin = new JButton("Sin");
this.add(btnSin, new GridBagConstraints(1, 6, 1, 1, 0.0, 0.0,
   GridBagConstraints.CENTER, GridBagConstraints.NONE,
   new Insets(2, 2, 2, 2), 0, 0));
```

Note: its better using `CalculatorPanel.CalculatorButton` instead of usual `JButton` as it provides most of necessary initializations.

Assigning Key Strokes

It's possible to assign up to three particular keystrokes per each calculator's button. Assigned keystokes will allow the user to execute Calculator operation using keyboard's key and/or key combinations. For tour new "sine" operation we will assign Ctrl+1 keystroke.

```
btnSin.putClientProperty(CalculatorPanel.BUTTON_KEYSTROKE_1,
                         KeyStroke.getKeyStroke(KeyEvent.VK_1,
                         KeyEvent.CTRL_MASK));
```

Register sine operation in the Calculator

```
this.getCalculator().addOperation(btnSin.getActionCommand(),
  new SinCalculatorOperation());
```

That is it!

Now it's possible to add ExtCalculatorPanel component to the application, compile and use calculator with additional sine operation.

Feedback

As part of the continuing effort to improve our product, we welcome your comments, suggestions and general feedback regarding the product.

If you have questions about Components!, please feel free to contact us for further information at components@jproductivity.com or visit our site using the following URL: http://www.jproductivity.com.

If you discover any issues or defects in Components!, please send the description of them to components@jproductivity.com.