

8.09.0---FAQ

1.---Chartsequal

```
/**
 * Object
 * @param ob  Object
 * @return boolean
 */
public boolean equals(Object ob) {
    return ob instanceof Chart && super.equals(ob)
        && ComparatorUtils.equals(((PieChart) ob).getChartTitle(), chartTitle)
        && ComparatorUtils.equals(((PieChart) ob).getChartDescribe(), chartDescribe)
        && ComparatorUtils.equals(((PieChart) ob).getChartDescribe(), roma);
}
```

2.cannot be cast to **com.fr.plugin.basic.Closeable**---plugin.xml<main-package>

<http://dev.fanruan.com:8090/pages/viewpage.action?pagelId=5374139>

9.0xml9.0**com.fr.plugin8.0com.fr** plugin.xml<main-package>

```
<main-package>com.personal.main</main-package>
```

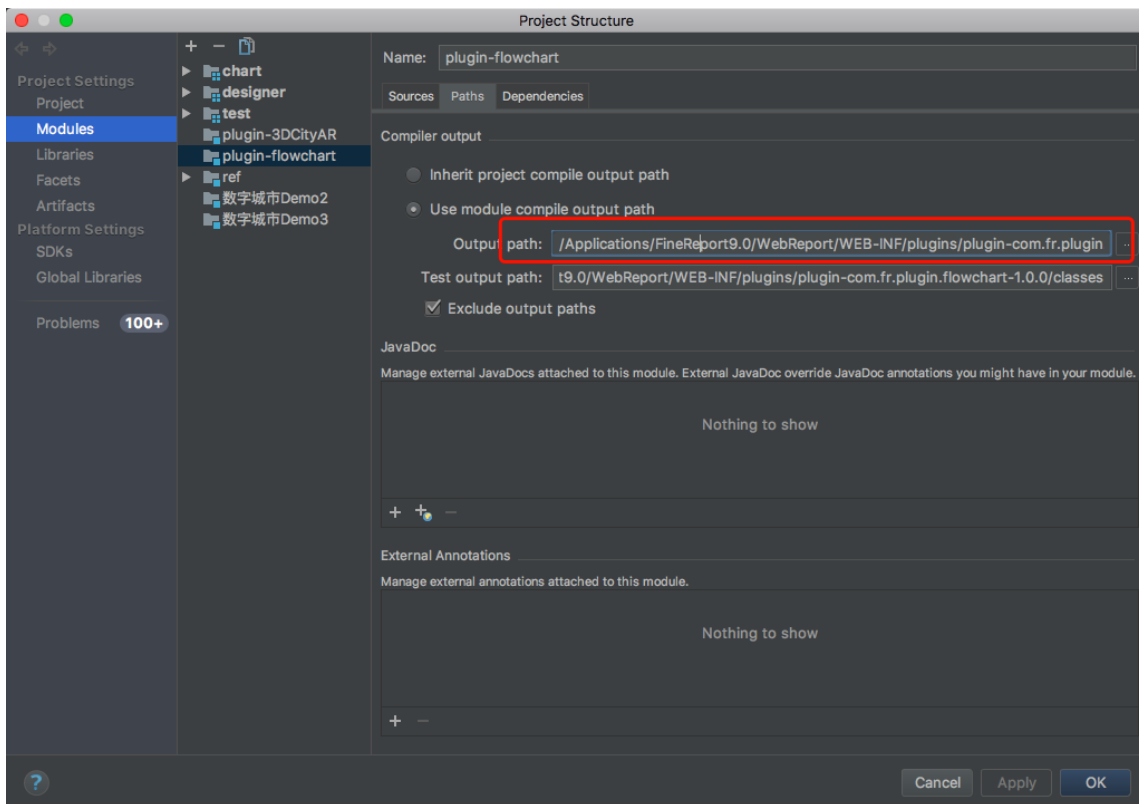
3.—ActualLocaleFinder

<http://dev.fanruan.com:8090/pages/viewpage.action?pagelId=327732>

4. java.lang.**ClassNotFoundException**:**com.fr.plugin.XXX**

<http://dev.fanruan.com:8090/pages/viewpage.action?pagelId=5373985>

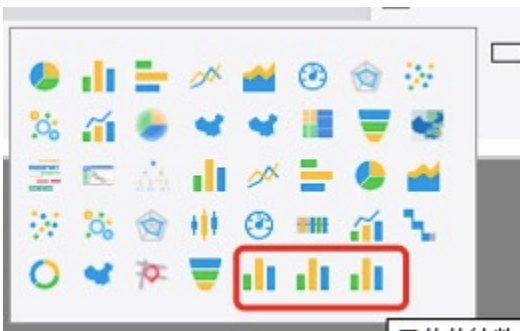
moudleclass-output path env/home class



fr9.0/Applications/FineReport9.0/Applications/FineReport9.0/WebReport/WEB-INF

output path /Applications/FineReport9.0/WebReport/WEB-INF/plugins/plugin-com.fr.plugin.XXX-1.0.0/classes

5.



AbstractIndependentChartsUIgetIconPath

```

@API(level = IndependentChartUIProvider.CURRENT_API_LEVEL)
public abstract class AbstractIndependentChartsUI extends AbstractProvider implements IndependentChartUIProvider {

    public String mark4Provider() { return getClass().getName(); }

    @Override
    public AbstractChartTypePane getPlotTypePane() { return new PiePlotPane(); }

    @Override
    public ChartDataPane getChartDataPane(AttributeChangeListener listener) { return new ChartDataPane(listener); }

    @Override
    public AbstractTableDataContentPane getTableDataSourcePane(Plot plot, ChartDataPane parent) {
        return new CategoryPlotTableDataContentPane(parent);
    }

    @Override
    public AbstractReportDataContentPane getReportDataSourcePane(Plot plot, ChartDataPane parent) {
        return new CategoryPlotReportDataContentPane(parent);
    }

    public BasicBeanPane<Plot> getPlotSeriesPane(ChartStylePane parent, Plot plot) { return null; }

    public BasicBeanPane<Plot> getPlotSeriesPane() { return null; }

    public boolean equals(Object obj) { return obj != null && ComparatorUtils.equals(obj.getClass(), this.getClass()); }

    public ConditionAttributesPane getPlotConditionPane(Plot plot) { return null; }

    @Override
    public AbstractChartAttrPane[] getAttrPaneArray(AttributeChangeListener listener) {
        return new AbstractChartAttrPane[]{};
    }

    @Override
    public boolean isUseDefaultPane() { return false; }

    @Override
    public String getIconPath() { return "com/fr/design/images/form/toolbar/ChartF-Column.png"; }
}

```

6.---BaseFormula

9.0BaseFormula8.0...so

```

Object object = "=now()";
CalculatorProvider calculator = CalculatorProviderContext.getCalculatorProvider();
FormulaProvider formula = new Formula((String) object);
StableUtils.dealFormulaValue(formula, calculator);
object = formula.getResult();

```

7.

1DemoReportDataContentPane AbstractReportDataContentPane

```

public class DemoReportDataContentPane extends AbstractReportDataContentPane {

    public DemoReportDataContentPane(ChartDataPane parent) {
        this.setLayout(new BorderLayout());
        this.add(new UILabel("", UILabel.CENTER), BorderLayout.CENTER);
    }

    @Override
    protected String[] columnNames() {
        return new String[0];
    }

    @Override
    public void populateBean(ChartCollection ob) {

    }
}

```

2AbstractIndependentChartsUI getReportDataSourcePane

```

/**
 * @param plot
 * @param parent
 * @return
 */
@Override
public AbstractReportDataContentPane getReportDataSourcePane(Plot plot, ChartDataPane parent) {
    return new DemoReportDataContentPane(parent);
}

```