

The Beauty of Charts and Graphs in BUI

From GUI to BUI: BBJChart

BBBj® implements a rich API for constructing graphical charts with the **BBjChart** control, which is built on top of the **JFreeChart** API. The BBJChart control presents a unique challenge for BUI – it exposes implementation details that depend on the assumption that the JFreeChart API is running in a Java Runtime Environment on the client. But BUI doesn't run in a client-side JRE; in fact, it runs on platforms (like iOS) that don't support Java at all.

To make **BBjBarChart**, **BBjLineChart** and **BBjPieChart** work in BUI, we moved the guts of the implementation from the client to the web server. We monitor all changes that you make to your chart, and every time we see a change or a batch of changes, followed by no changes for 250 milliseconds, we transparently generate an updated snapshot of the chart as an image and push that image over to the client. Using this strategy, most existing BBJChart programs automatically work in BUI as of BBj 11.10. **Figure 1** shows a BBJChart sample program running in GUI, and one in BUI on the desktop. Download and run this sample at links.basis.com/11samples.

In the BBJChart samples in shown **Figure 1**, you can see that only three of the four samples work in BUI. The fourth chart uses functionality that is not available in BUI. **BBjGenericChart**, **setClientChart** and **getClientChart** are not implemented in BUI because they depend on the infrastructure for Java client objects. But, with some minor adjustments, it is still possible to work with the full JFreeChart API in a BBJ program that will run in BUI. The solution is to use the same basic strategy that BBJ BUI itself uses – build the chart on the server, then push the

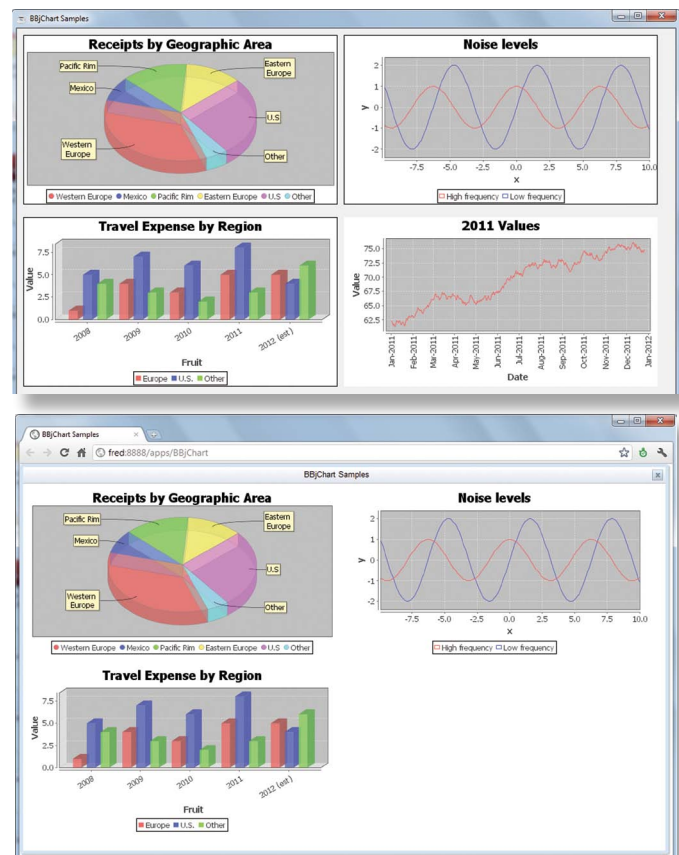


Figure 1. Four BBJCharts in GUI (top) and three charts in BUI desktop (bottom)

rendered chart over to the client as an image. **Figure 2** shows this conversion process for a representative **BBjGenericChart**.

Figures 3 shows the results of the converted JFreeChart sample program. Note that the sample program uses randomly generated data each time it's invoked so the resulting charts are not identical. Download and run this sample located at links.basis.com/11samples. >>



By Jim Douglas
Software Developer

BBJGenericChart

```

series! = new TimeSeries("Random Data")
current! = new org.jfree.data.time.Day@1(1,1,2011)
value = 62
for i = 0 to 360
    delta = i/60
    value = value + Math.random() - .5
    series!.add(current!,new Double(value + delta))
    current! = current!.next()
next i
dataset! = new TimeSeriesCollection@!(series!)

chart! = ChartFactory@.createTimeSeriesChart("2011
Values", "Date", "Value", dataset!, 0, 0, 0)

list! = bbjapi().makeVector()
list!.add(new Float(10).floatValue())
list!.add(new Float(1).floatValue())
CAP_ROUND = BasicStroke.CAP_ROUND
JOIN_ROUND = BasicStroke.JOIN_ROUND
stroke! = new
BasicStroke(1,CAP_ROUND,JOIN_ROUND,5,list!,0)

renderer! = new XYLineAndShapeRenderer()
renderer!.setBaseShapesVisible(0)
renderer!.setSeriesStroke(0,stroke!)
plot! = chart!.getPlot()
plot!.setRenderer(renderer!)

format! = new SimpleDateFormat("MMM-yyyy")
tickUnit! = new
DateTickUnit(DateTickUnit.MONTH,1,format!)
axis! = plot!.getDomainAxis()
axis!.setTickUnit(tickUnit!)
axis!.setVerticalTickLabels(1)

```

```

genericChart! =
window!.addGenericChart(104,520,300,500,270)
genericChart
genericChart
genericChart
.setMouseZoom

```

```

genericChart! =
window!.addGenericChart(104,520,300,500,270)
genericChart!.setToolTipText("BBJGenericChart")
genericChart!.setClientChart(chart!)
genericChart!.getClientChartPanel()
.setMouseZoomable(1,0)

```

JFreeChart

```

series! = new TimeSeries("Random Data")
current! = new org.jfree.data.time.Day(1,1,2011)
value = 62
for i = 0 to 360
    delta = i/60
    value = value + Math.random() - .5
    series!.add(current!,new Double(value + delta))
    current! = current!.next()
next i
dataset! = new TimeSeriesCollection(series!)

chart! = ChartFactory.createTimeSeriesChart("2011
Values", "Date", "Value", dataset!, 0, 0, 0)

list! = bbjapi().makeVector()
list!.add(new Float(10).floatValue())
list!.add(new Float(1).floatValue())
CAP_ROUND = BasicStroke.CAP_ROUND
JOIN_ROUND = BasicStroke.JOIN_ROUND
stroke! = new
BasicStroke(1,CAP_ROUND,JOIN_ROUND,5,list!,0)

renderer! = new XYLineAndShapeRenderer()
renderer!.setBaseShapesVisible(0)
renderer!.setSeriesStroke(0,stroke!)
plot! = chart!.getPlot()
plot!.setRenderer(renderer!)

format! = new SimpleDateFormat("MMM-yyyy")
tickUnit! = new
DateTickUnit(DateTickUnit.MONTH,1,format!)
axis! = plot!.getDomainAxis()
axis!.setTickUnit(tickUnit!)
axis!.setVerticalTickLabels(1)

```

```
out! = new ByteArrayOutputStream()
```

```

out! = new ByteArrayOutputStream()
ChartUtilities.writeChartAsPNG(out!, chart!, 500, 270)
bbjimage! =
imageManager!.loadImageFromBytes(out!.toByteArray())
chartControl! =
window!.addImageCtrl(104,510,290,500,270,bbjimage!)
chartControl!.setToolTipText("JFreeChart
TimeSeriesChart")

```

Figure 2. Converting the fourth chart, BBJGenericChart, to a JFreeChart for BUI presentation



Figure 3. All four BBJCharts with JFreeChart in BUI desktop and BUI iPad

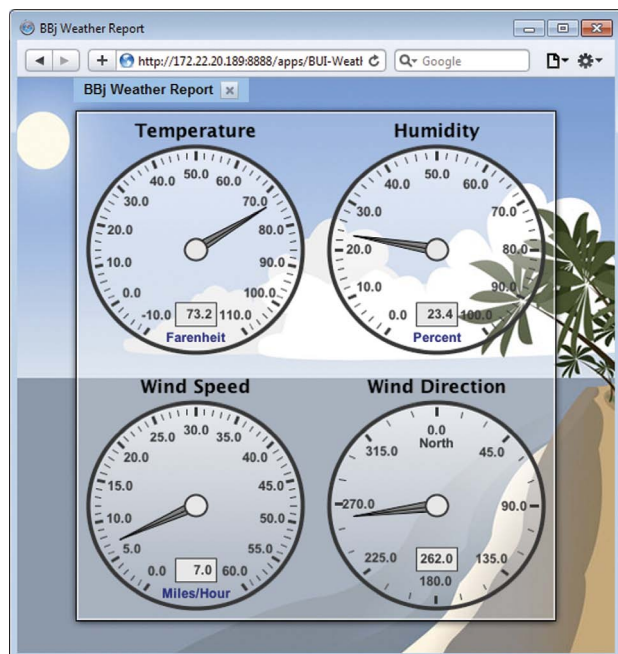


Figure 4. A BUI program using JFreeChart DialPlots

Applying those same kinds of changes to other JFreeChart-based programs works equally well.

Figure 4 shows the result of that modification to another program that uses JFreeChart DialPlots to show current weather conditions.

As BASIS' BUI technology matures, more and more of the traditional GUI functionality becomes available for use in the browser. Common controls such as buttons, listboxes, and input controls have been running well in BUI from the start, however, a few stragglers such as BBjCharts have not been available due to complexities with their underlying architecture. BASIS continues to chip away at the last few bits of GUI functionality not yet implemented in BUI; BBjCharts are the latest set of controls that BASIS recently modified to function in the BUI environment. Updates and enhancements are sure to come, so follow the ongoing progress in the "What's Implemented?" and "What's Not Yet Implemented?" lists in the [BBj BUI: Getting Started](#) document. ■



- For more information, refer to the online [BBj*Chart documentation](#)
- For features needed in the field that are not yet implemented in BUI, post a note on the BBj-Developer list at groups.google.com/group/bbj-developer/topics; subscribe at www.basis.com/discussion-forums

Partnership

AddonSoftware Steps Into the Cloud

This year's TechCon "tour," *Step into the Cloud with BASIS*, informed audiences around the world in Las Vegas, Nevada; Montreal, Canada; and Saarbrücken, Germany, of BASIS' own move to the cloud. Nearly all of BASIS' business functions – their custom version of the AddonSoftware® ERP solution, source code archives, build processes, test machines as well as company email and documentation – all now run in the cloud in data centers scattered across the United States and provide a local cloud frontage presence around the globe.

Introducing...

Using first-hand cloud experience as the template, BASIS introduces AddonSoftware Cloud Services, the software-as-a-service (SaaS) version of AddonSoftware by Barista®. This new service presents BASIS partners with a quicker and simpler way to deploy the AddonSoftware Enterprise Resource Planning (ERP) solution. BASIS prides itself on offering choices and flexibility to its partner community and is happy to extend this core value. Now AddonSoftware can run on-premise in the traditional way or online through an external host in a cloud environment...a single solution with deployment options to meet the business needs of developers and end users alike.

Strategically, the new cloud offering makes AddonSoftware more affordable to use, own, and deploy as illustrated in **Figure 1**. Affordability is achieved both through the low user-based monthly subscription model and the elimination of the typically large capital outlays for hardware, installation, and maintenance. Installations are quicker and there is no more guessing involved with purchasing a server that could be under- or over-sized. With the cloud, you can upgrade or downgrade at a moment's notice. Similarly, there is no concern that you'll be stuck running on last year's model; [Moore's Law](#) keeps you on the forefront of technology without the cost of continuous hardware investments. The immediate loss-of-value of purchasing a new car doesn't hold a candle to the immediate loss-of-value of purchasing computer hardware. The cloud is an infrastructure extension of the BASIS building blocks concept whereby software developers can remain focused on developing their vertical specialties while enjoying the benefits of a hosted solution. >>



By Paul Yeomans
Vertical Market
Account Manager