Dash Boredom With the Dashboard Utility

By Nick Decker
Engineering Supervisor

Following the release of the Dashboard Utility in BBj® 14.0, BASIS introduced this utility in a few Java Breaks and published several live demos on the BUI Showcase page. To recap, this article covers some of the Dashboard Utility's more prominent features and capabilities, and provides images and screenshots to give you a better feel for all it has to offer.

Dashboards – A Blast From the Past

At BASIS TechCon in 2007, we presented a GUIBuilder-built demonstration program called DigitalDashboard. BBjCharts were new to the language at the time, and the program's purpose was to illustrate how to utilize those charts to show the Chile Company's sales team's performance.

Fast forward several years to the release of BBj 14.0 that now comes bundled with the Dashboard Utility. The underlying technology of the Dashboard Utility is the same as that of the BBjChart methods, but the differences are like night and day.

- BBjCharts are independent, low-level controls that you add to your BBj program and manage with your code.
- The Dashboard Utility is an advanced framework built upon the foundation of a Dashboard Widget, which includes multiple types of charts and also supports grids, reports, images, and HTML.
- The utility provides built-in widget management so that it will take care of categorizing, sizing, positioning, hiding or showing, and even refreshing your widgets for you.

After comparing the original TechCon demo with a couple of BUI dashboard apps running in iPads shown in **Figure 1**, the old slogan "*You've come a long way, baby!*" comes to mind.

Why Dashboards are so Effective

The core concept of a dashboard is to take complicated information and present it in a simple visual format. This makes the data significantly easier to understand, allowing you to grasp several performance metrics at a glance. Charts

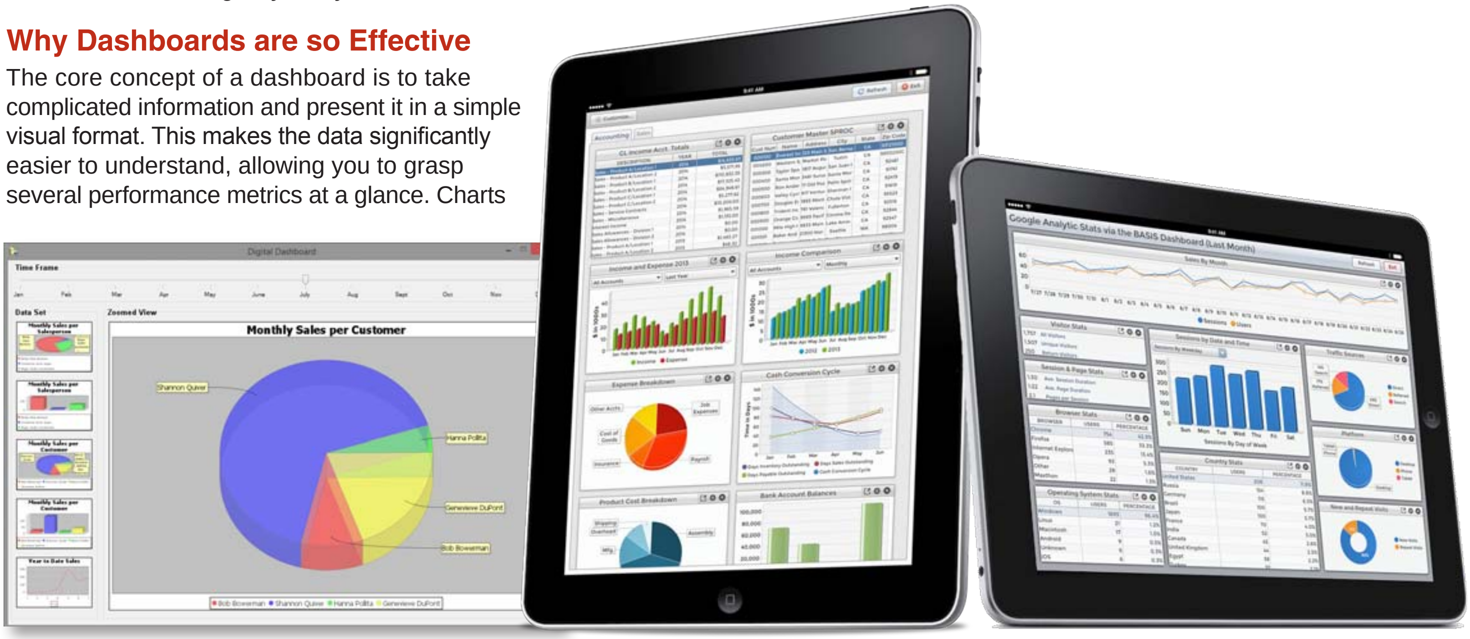


Figure 1. The 2007 Digital Dashboard demo (left) compared to the contemporary dashboard BUI apps running on iPads (right)

are particularly effective, as they exploit our ability to correlate visually spatial locations and distances, sizes, and colors. These fuse together and leave us with a distinct notion of data relationships in a way that is immediately obvious – something rows and columns of numbers often cannot do. In so doing, owners and executives are empowered to make faster and more informed decisions to make their businesses better. **Figure 2** shows three different dashboard chart widgets that demonstrate data correlation.

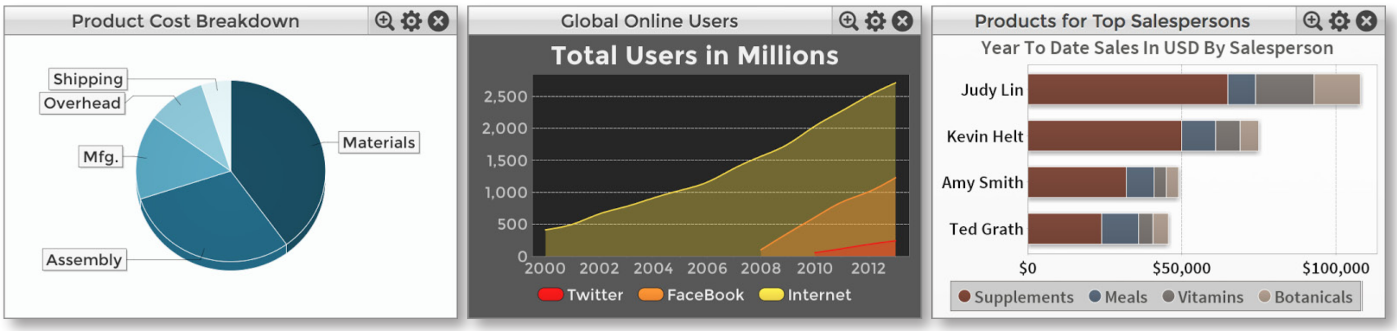


Figure 2. Dashboard widgets demonstrating data correlation in pie, line, and stacked bar charts

The pie chart shown on the left gives a clear relationship of the relative amounts a business spends on product costs based on the size of the pie slice. The line chart shown in the center visually indicates the increasing number of online users as the years go by. The slopes of the lines in the chart also convey each of their relative growth rates. The stacked bar chart shown on the right provides several points of interest. The length of the bars and the ordering of the sales reps make it simple to discern their relative performance over the past year. Looking more closely at each sales rep's bar also presents us with a breakdown of their yearly sales by product. In each one of these cases, our dashboard charts effectively present complex trend, cost, and sales information in a readily consumable format.

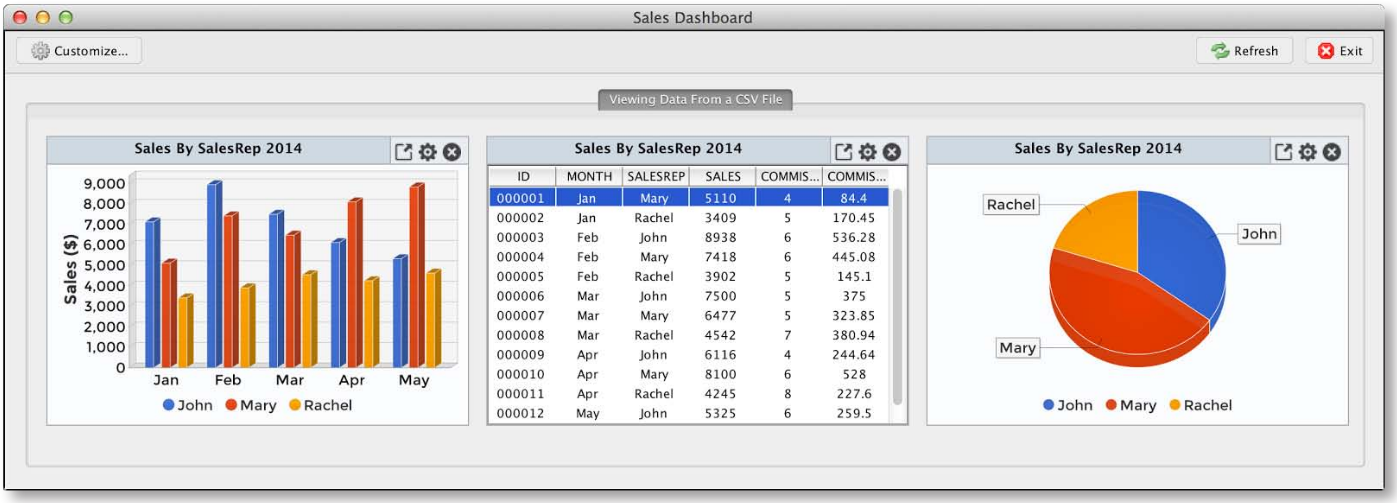
The Dashboard Utility's Goals

When BASIS engineers devised the Dashboard Utility, two leading tenets guided the design philosophy.

1. Developers should be able to get a dashboard with widgets running without a lot of code.
2. Widgets should be customizable, but they ought to look good by default.

How well did we do on each of those? As a testament to the former, our Adding the New Digital Dashboard to Your App Java Break video not only introduces the utility, but gives a complete walk-through of the dashboard creation process. During the video, we show the code we used to create a dashboard with a grid, bar, and pie chart that all report on different facets of sales data that was exported from a spreadsheet. The YouTube page (links.basis.com/youtube) includes a link to the final source code that is just over 30 lines, not counting the REM and empty lines that exist to aid in legibility.

The resultant dashboard shown in **Figure 3** confirms that we succeeded in our first goal. In just over a couple dozen lines of code, we have a fully functional dashboard up and running.



ID	MONTH	SALESREP	SALES	COMMIS...	COMMIS...
000001	Jan	Mary	5110	4	84.4
000002	Jan	Rachel	3409	5	170.45
000003	Feb	John	8938	6	536.28
000004	Feb	Mary	7418	6	445.08
000005	Feb	Rachel	3902	5	145.1
000006	Mar	John	7500	5	375
000007	Mar	Mary	6477	5	323.85
000008	Mar	Rachel	4542	7	380.94
000009	Apr	John	6116	4	244.64
000010	Apr	Mary	8100	6	528
000011	Apr	Rachel	4245	8	227.6
000012	May	John	5325	6	259.5

Figure 3. The dashboard program built during the Java Break

What do we mean by 'fully functional'? When you resize the window, the widgets will automatically resize and reposition themselves to maximize their use of the available window space. You can also reposition the widgets, hide and show them, pop them out to view a larger version; save out an image of the chart, email it to a colleague; refresh the widget to reflect the latest data, and more.

For our second goal, we modified dozens of chart parameters so that the default widgets look terrific without any extra work on your part. To see the difference, it's worth taking another look at the comparison of charts shown in **Figure 1**. The charts in the Dashboard

Utility are brighter, cleaner, and have improved fonts and colors that produce a more aesthetic and professional result. You still have access to scads of chart methods that offer greater control over several components including data colors and font family/size/color, but those are added enhancements instead of necessary intricacy.

The Dashboard Utility's Features

The Dashboard Utility is split between two BBj object-oriented programs that together offer over 750 methods, so it's safe to say that the Dashboard Utility is a fairly advanced library. The list of features and capabilities is not only extensive, but has been steadily growing over the last several months since its introduction. Rather than providing an exhaustive list of every option and available method, let us take a whirlwind tour of several of the Dashboard Utility's more prominent features.

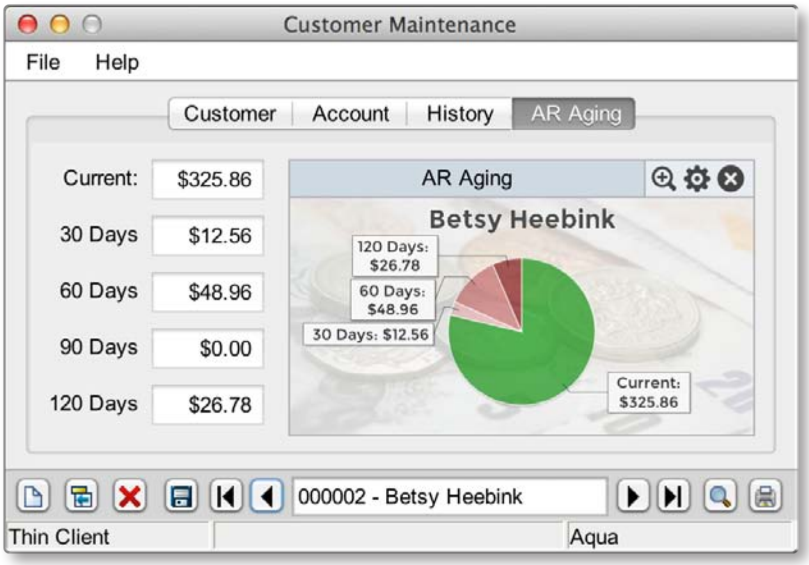
Dashboard Features

- **Dashboard Modes**. You can run a full-featured dashboard as a stand-alone program in a top-level window, or create it inside a BBj control to embed in your application as shown in **Figure 4**. That same flexibility applies to dashboard categories (tabs) and widgets as well. This makes it possible for you to embed any portion of a dashboard into a new or existing BBj program, or run them in a stand-alone fashion.

- **Dashboard Categories.** You can create multiple categories for a dashboard such as shown in **Figure 5**. The dashboard displays these categories in separate tabs, making it easy to provide multiple dashboard widgets without overcrowding the window or overwhelming the user.

- **Widget Size.** The dashboard resizes and repositions its widgets automatically in response to resizing the dashboard window or the browser window running in BUI. You can influence the size of the widgets by setting a minimum and maximum width, and you can also set the column and row spacing. This is done on a per-category basis, and makes it possible to have larger widgets for displaying reports and smaller widgets for displaying data grids. You can also modify the widget's size and spacing to target particular screen resolutions when running on a mobile device.

- **Widget Placement.** You can reorganize widgets in a dashboard category by simply dragging a widget via its title bar (see **Figure 6)** and dropping it into the desired position. The dashboard rearranges the other widgets automatically to make room for the newly positioned widget. The dashboard stores the position for each widget in a cookie so after you arrange the widgets, it will remember their order and display them in the same position the next time you run the dashboard.



Current:	$325.86
30 Days	$12.56
60 Days	$48.96
90 Days	$0.00
120 Days	$26.78

Figure 4. A pie chart widget embedded in an AppBuilder-generated application

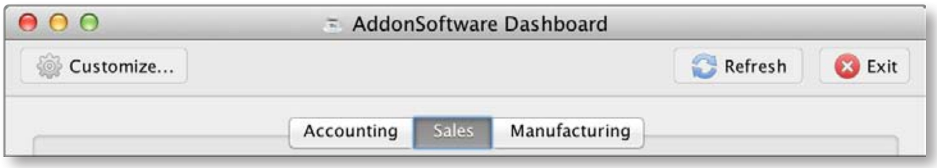


Figure 5. Examples of different categories in the AddonSoftware Dashboard

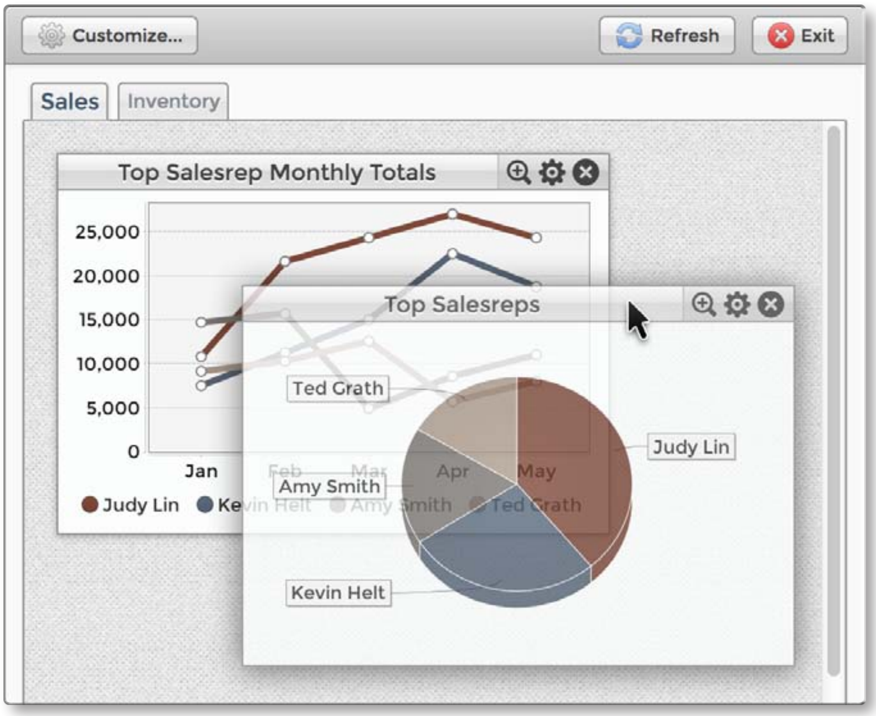


Figure 6. Dragging a widget to reposition it in the dashboard

- **Widget Control.** Dashboard widgets all have a toolbar on the top of the widget, as shown in **Figure 7**, that allows you to 'pop out' the widget, configure the widget, or close the widget. Closing a widget in the dashboard hides it so that it no longer appears in the dashboard. You can click on the dashboard's [Customize] button to see the full list of available widgets for the current dashboard category, along with a screenshot and brief description of each widget. The 'Customize' window makes it easy to add a hidden widget back to the dashboard. The dashboard also stores the visible/hidden state of the widgets in a cookie, so if you hide a widget then the dashboard will hide it the next time you run the program as well.

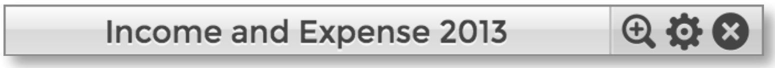
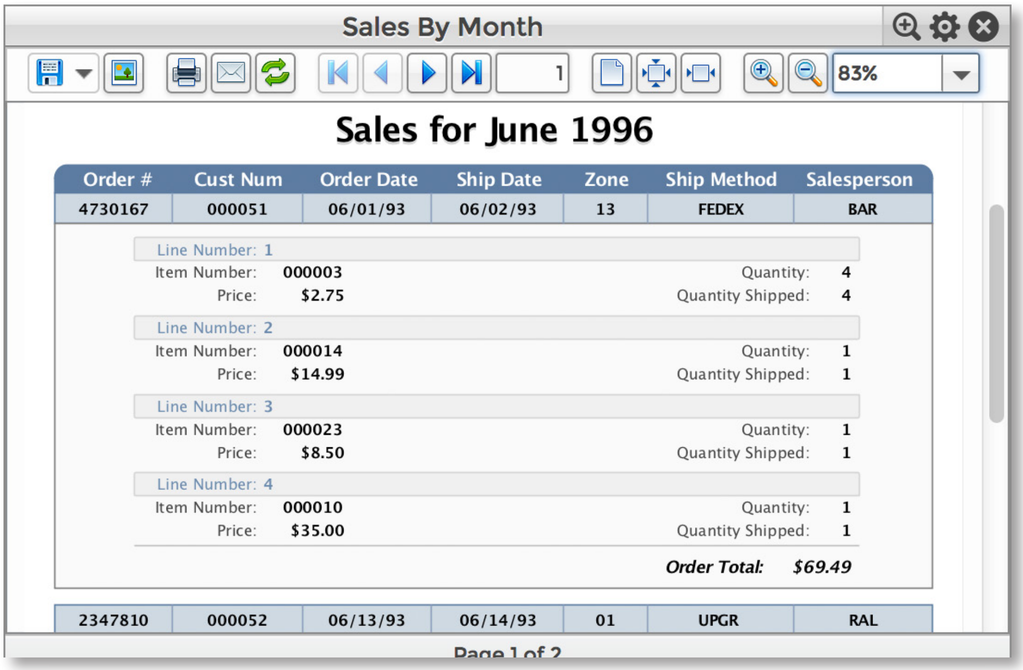


Figure 7. A sample widget's toolbar with controls on the right side

Widget Features

- **Widget Types.** The dashboard supports several different types of widgets, allowing you to display data in a variety of different formats. Charts are commonly used in dashboards, but you may also display grids, images, web pages/other HTML content, and even full-featured reports via BBJasper as shown in **Figure 8**. Various chart types are supported as well, including bar charts, stacked bar charts, line charts, area charts, stacked area charts, pie charts, ring charts, and more.



Sales for June 1996

Order #	Cust Num	Order Date	Ship Date	Zone	Ship Method	Salesperson
4730167	000051	06/01/93	06/02/93	13	FEDEX	BAR

Line Number: 1
Item Number: 000003 Quantity: 4
Price: $2.75 Quantity Shipped: 4

Line Number: 2
Item Number: 000014 Quantity: 1
Price: $14.99 Quantity Shipped: 1

Line Number: 3
Item Number: 000023 Quantity: 1
Price: $8.50 Quantity Shipped: 1

Line Number: 4
Item Number: 000010 Quantity: 1
Price: $35.00 Quantity Shipped: 1

Order Total: *$69.49*

Order #	Cust Num	Order Date	Ship Date	Zone	Ship Method	Salesperson
2347810	000052	06/13/93	06/14/93	01	UPGR	RAL

Figure 8. A JasperViewerWidget displaying a report in a dashboard widget

- **Widget Data.** You can create many dashboard widgets with an SQL connect string and query to automatically build a dataset and populate the widget. When you create a widget in this manner, it is automatically refreshable and will respond to a refresh event by requerying the database and updating the widget with the latest information.

- **Widget Refresh.** You can configure refreshable dashboard widgets programmatically to refresh themselves automatically, and users may accomplish the same via the widget's configuration menu (**Figure 9**). The widget will then update itself with new data whenever the specified refresh interval time has elapsed. You can configure the interval by providing the desired number of seconds, minutes, or hours that must elapse before the widget refreshes itself. The dashboard stores the automatic/manual refresh configuration in a cookie, so when you set a widget to refresh itself automatically it will continue to do so the next time you run the dashboard.

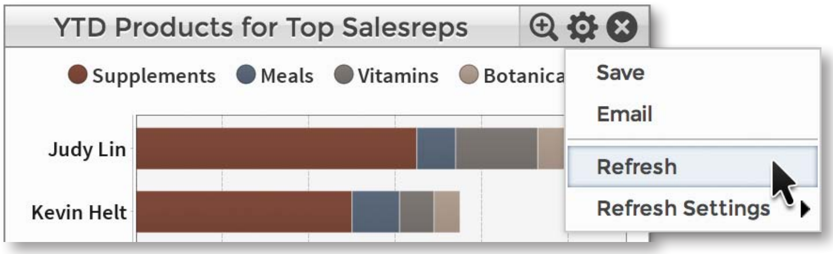


Figure 9. Manually refreshing the widget from the toolbar menu

- **Widget Filters and Links.** Dashboards widgets also support optional filters and links. Filters are displayed as dropdown lists at the top of the widget and allow the end user to modify the contents of the widget. A common use case for filters is to modify the underlying data query so that you can select values like 'Domestic', 'Overseas', etc. to change the reporting range for the widget (see **Figure 10**). Links are located on the bottom of the widget and provide an easy way to open up a web page or run a BBj program of your choice.

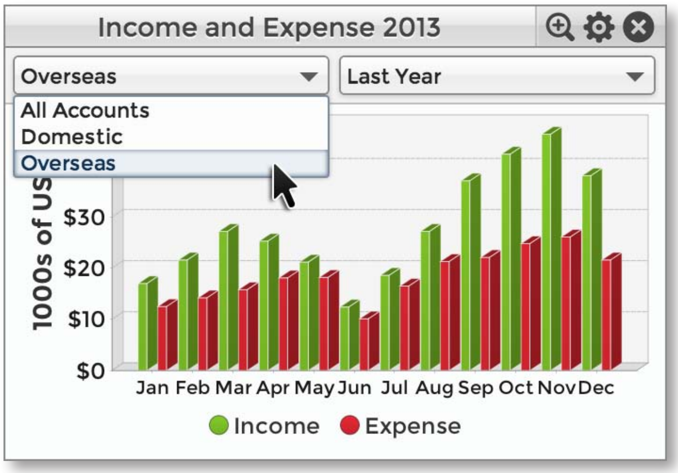


Figure 10. Accessing a widget's filter to select the accounts on which to report

Utility Features

- **CSS Customizability.** The dashboard and widgets offer dozens of CSS selectors so you have complete control over the look and feel of your dashboard running in BUI. For example, a dashboard category will have the general dashboardCategoryWindow selector as well as a selector based off of the name you gave the category such as dashboardCategorySalesReports. This way you can style all of the categories and widgets at once or style each one individually with a different appearance. See the example of custom colors and CSS in **Figure 11**.

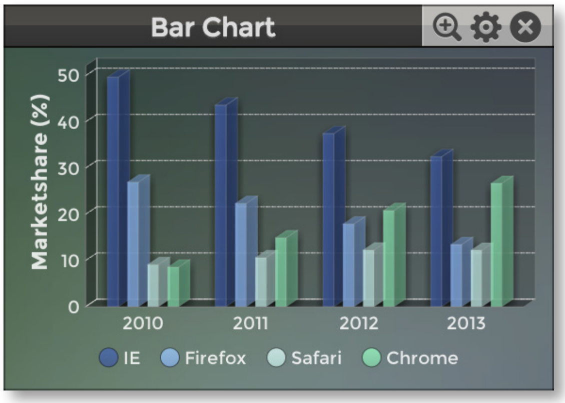


Figure 11. A chart widget with custom colors and CSS

- **High Pixel Density Display.** The dashboard program and the widgets it creates take advantage of high pixel density displays when available, such as on Apple Retina devices. The fonts, charts, and even the widget and button icons were optimized to run at full resolution with more detailed graphics on a high pixel density display – both in the traditional thin client and in BUI. **Figure 12** shows a comparison of a normal display and the high density display.

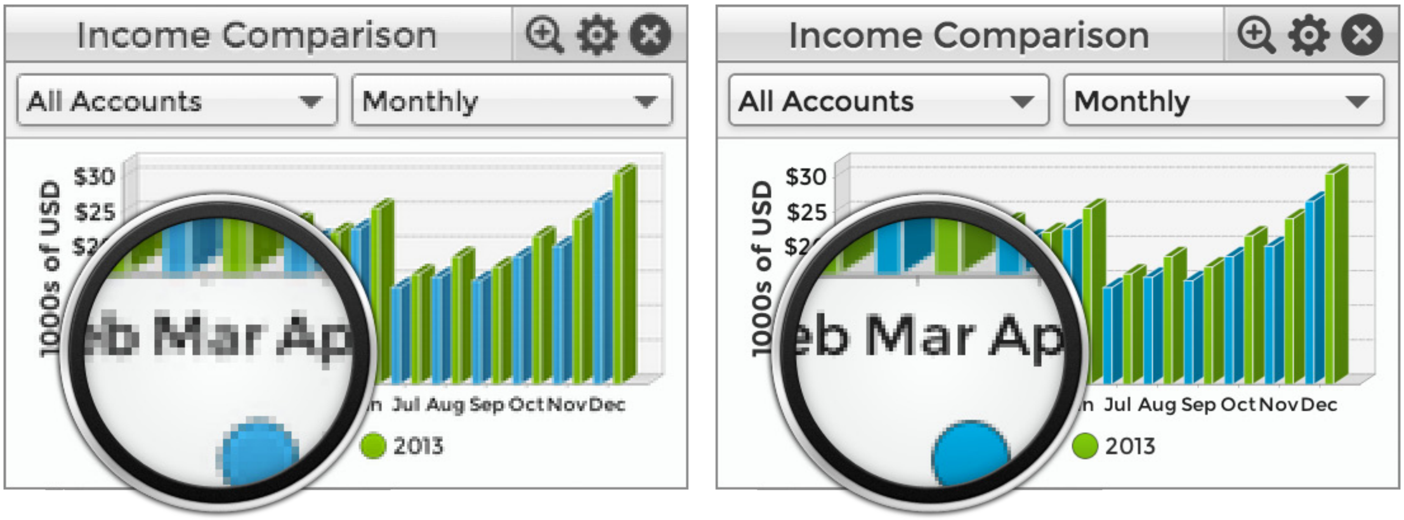


Figure 12. The dashboard on a normal display (left) and a high density display (right) that has quadruple the resolution and increased sharpness

Summary

While BBjCharts have been available in the language for several years, building a dashboard using these charts has always taken quite a bit of time, effort, and code. The new BASIS Dashboard Utility makes all of that a thing of the past. By providing extensive built-in functionality and over a dozen different types of widgets, you can now get a fully functional dashboard up and running in a couple dozen lines of code. Better yet, it handles everything from sizing and positioning widgets to automatic refresh and save functionality. By taking advantage of all that the Dashboard Utility has to offer, you can equip your customers with the ability to make better, faster business decisions with a powerful and enlightening view into their data, all with very little effort on your part! ■

- Read these articles also appearing in this issue
  - *Easier Decision Making With the Dashboard Utility*
  - *The Magic of the Widget Wizard*
- Refer to online documentation
  - *Dashboard Utility Overview*
  - *JasperViewerWidget*
- Watch the Java Break *Adding the New Digital Dashboard to Your App* and our other YouTube videos