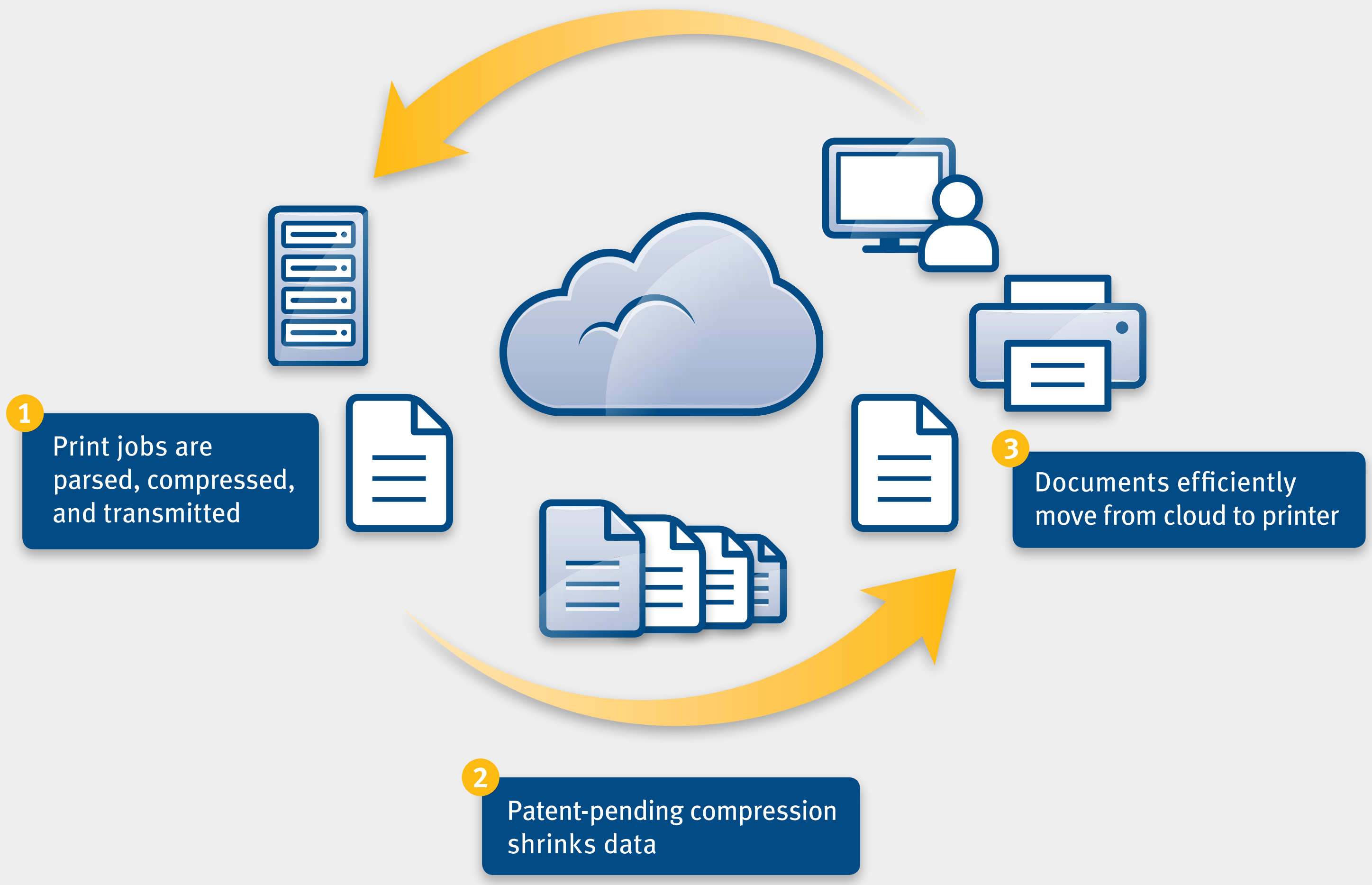
BASIS International Advantage®

Volume 17 • December 2013

CirrusPrint™
Cloud and Network Printing


1
Print jobs are parsed, compressed, and transmitted
2
Patent-pending compression shrinks data
3
Documents efficiently move from cloud to printer

Features

- High performance wide-area network printing
- Fast, efficient multi-location file transfer
- Very high compression ratios, up to 98% through patent-pending anti-redundancy techniques, even of already compressed PDF documents
- One to one, or one to many, document distribution features: one file to many desktops, for example
- Secure document transmission using SSL
- Stateless design, so document transmission can occur as remote sites are available
- Exact replication of print jobs retains features such as duplex and tray control
- Server-centric configuration via browser interface
- Reduced bandwidth utilization

SDSI

Synergetic Data Systems, Inc.
Intelligent Tools for Unix®, Linux®, Windows®

CirrusPrint.com

BUI, BUI Everywhere

Our "BUI Everywhere" cover makes no small claim...it hardly seems possible, but it is true. Today, you can take your graphical BBx® application, written in Visual PRO/5® or BBj® – the latest version of BBx – and, *without any code changes* and *with zero deployment*, run it from a desktop web browser or from your mobile device of choice *without any Java or BASIS component* on the desktop or mobile device! BUI is phenomenal, truly amazing, and best of all, you don't have to take our word for it, you can try it for yourself!

From www.basis.com, just navigate to the top left menu bar immediately below the BASIS logo to [Showcase] and then to [BBx Web Showcase]. Or go to links.basis.com/buidemos. Try one of our sample programs from your favorite tablet, smartphone, or modern desktop browser and you'll be running a BBj program hosted from an Amazon Cloud server nearest you! How do we do it? At the point where we interpret BBx code into Java code, we can now also interpret your BBx code into JavaScript and HTML5, technologies that are embedded into, and are integral parts of, most any modern browser; mobile, or desktop.

In this issue, we highlight articles that show off new BUI feature and function, and discuss why BASIS' BUI is the perfect choice for developing mobile data-driven business applications. We look forward to hearing of your first, or your very latest, successful deployment of your own BUI application. Call your BASIS Account Manager and we can help you publicize your efforts to maximize the return on your development investment. We set a new record in this issue for jam-packing it with articles covering topics that explore new BASIS product features and function, performance improvements, and end-user success stories.

BASIS travelled the world both physically with our TechCon and TechCon*densed* events, and virtually to your office or home with ten new and several updated Java Breaks during the past twelve months. If you missed any, view them on our BASIS YouTube channel at www.youtube.com/BASIS (over 6,500 views in the past year) or from our website.

Speaking of the virtual world, we host active forums to facilitate ongoing peer-to-peer technical conversations; bbx-list, bbj-developer, barista-list, and addon-developer. All forums are open to BASIS partners and end users, except the addon-developer forum that we reserve for the AddonSoftware co-op developer community. We look forward to interacting with you on the forums or via Java Breaks. See you online! ■

Nico Spence
Chairman & CEO

Editor in Chief *Nico Spence*
nspence@basis.com

Editor *Susan Darling*
sdarling@basis.com

Technical Editors *Dr. Kevin King, Nick Decker*
kking@basis.com, ndecker@basis.com

Art Director, Graphics *Patricia Catlett*
pcatlett@basis.com

Electronic Production *Amer Child*
achild@basis.com

The *BASIS International Advantage* magazine is published and distributed by BASIS International Ltd.

BASIS does not endorse any products mentioned in the *BASIS International Advantage* other than those products licensed from BASIS International Ltd.

The trademarks and registered trademarks owned by BASIS International Ltd. in the United States and other countries are listed at www.basis.com/trademarks

All other product and brand names are trademarks or registered trademarks of their respective companies.

Subscribe at links.basis.com/subscribe

BASIS International Ltd.
5901 Jefferson Street NE
Albuquerque, NM 87109-3432

Phone +1.505.345.5232
US Sales 1.800.423.1394
International +1.505.338.4188
www.basis.com info@basis.com

©2013 by BASIS International Ltd.

Stock photo credits: www.123rf.com and www.istockphoto.com

Table of Contents

3D Tek
Innovative Search Solutions

Finding the perfect match is an art, not an accident.

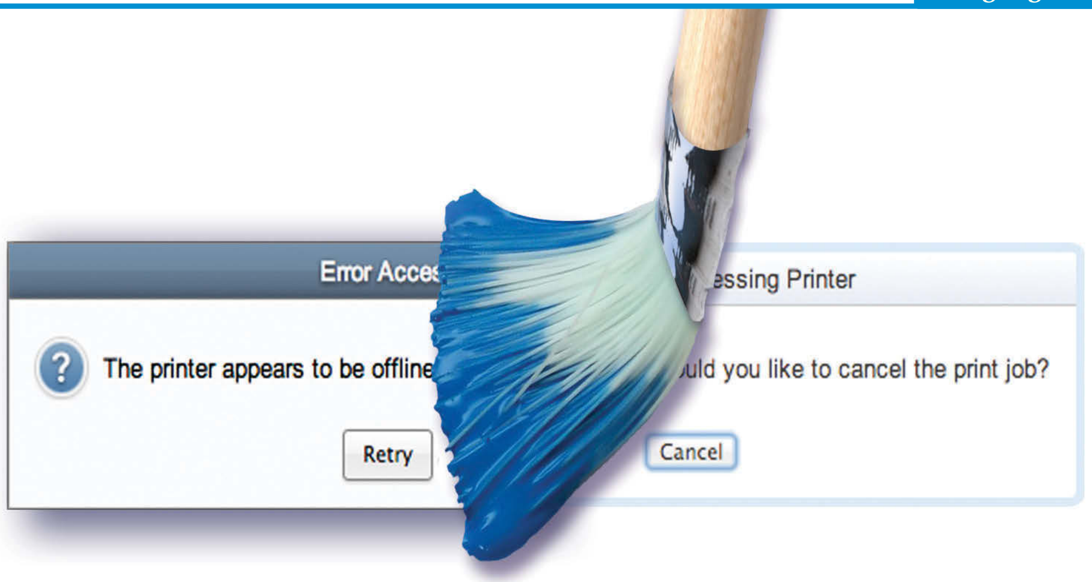
Technology is constantly evolving at BASIS International, and along with it, so are your staffing needs for BBj, VPRO/5 or Barista professionals. At 3D Tek we are dedicated to helping you maintain your competitive edge by finding the right professionals for your company.

Our custom search solutions go beyond conventional recruiting, allowing us to locate, recruit, and bring employees on board who not only have the talent and skill set you need, but who share your goals and reflect your company culture.

Whether you need someone for a contract, contract-to-hire, direct hire, or a fixed price project, we can find the perfect match.

Improve your productivity and profitability with 3D Tek, your IT & Executive search and recruitment partner.

352-569-9203 or visit us at www.3dtek.com


Error Accessing Printer
The printer appears to be offline. Would you like to cancel the print job?
Retry
Cancel

Default CSS Gets a Makeover

How would you like to instantly improve your BBj® web browser user interface (BUI) applications?

What about updating them to a contemporary look that improves the responsiveness of controls to give the user instant visual feedback while enhancing the user experience by making controls easier to click, select, highlight, spin, and navigate?

Although it sounds almost too good to be true, you will reap all of these rewards by simply updating to BBj 14.0, previewed in 13.10, to employ the new default BUI cascading style sheets (CSS) theme.

Why BASIS Updated the Default Theme

BBj BUI has always relied on CSS, utilizing the base GWT CSS file along with BASIS-custom definitions to control how a BBj application was rendered in the desktop or mobile browser. The default GWT theme was rather plain, painting splashes of bright blue here and there amid a sea of white to mimic more closely a basic web page. While perfectly functional, the theme started to show its age and struggled to compete with present-day web pages, web app solutions, and especially, mobile applications. Additionally, it did not exploit the full power of CSS to add user feedback to the interface, such as modifying a control's appearance when the user hovered over or clicked a button, gave focus to an input field, or tried to interact with a read-only control.

The original GWT CSS, created years ago, had to work consistently across all browsers available at the time, including Internet Explorer 6. Because of this requirement, it often catered to the lowest common denominator, foregoing useful functionality in the process. BASIS wanted any BUI app, from the simplest program to the most complex application, to look better and provide a more gratifying user experience. If the user runs a BUI app in a modern and capable browser, they should be able to take full advantage of its capabilities. Providing a new default CSS implementation was a great way to accomplish this as all BUI applications benefit from the new theme without any changes to configuration or code.

By Nick Decker
Engineering Supervisor

Improved Appearance

The new BUI default theme improves the general appearance of all BUI applications without any extra work from the developer. The theme revamps all BBj controls, updating their appearance from what can only be described as plain, to a more engaging and compelling presentation. The addition of subtle gradients, drop shadows, outlines, and colors goes a long way to provide a fresh look that is more pleasing, improves interaction, and reduces eye strain over long periods of use. BASIS applied as many of the thematic elements across multiple controls as applicable, providing a unified consistency amongst the components of a BUI application. Controls such as the BBjGrid and BBjTabCtrl now sit comfortably alongside every other BBjControl, with matching styles and colors.

Improved Interactivity

Although one of the primary goals was to improve the general appearance of any BUI application, this CSS project had a few secondary goals as well. For starters, the new theme was meant to be more interactive. Instead of changing a 1-pixel border around a hovered button from medium gray to medium blue, the new theme changes the appearance of the button by adding color and saturation to the entire control.

Another example of improved visual feedback occurs when you give an input control focus. Using the previous theme, giving an input control focus by clicking or tapping in it did not change the appearance of the control at all (depending on which browser you were using). The addition of a 1-pixel wide I-beam cursor was the only indication that the input control now had keyboard focus. This made it especially difficult to tell at a glance which control was in focus and was especially problematic when the user shifted focus to the control via the tab key or task-switched away from and back to the application. The new theme shown in **Figure 1**. adds a bright outline and an outer glow to a focused control, making it easier to distinguish it from the other input controls, and improves the contrast and legibility of text in read-only and disabled controls.

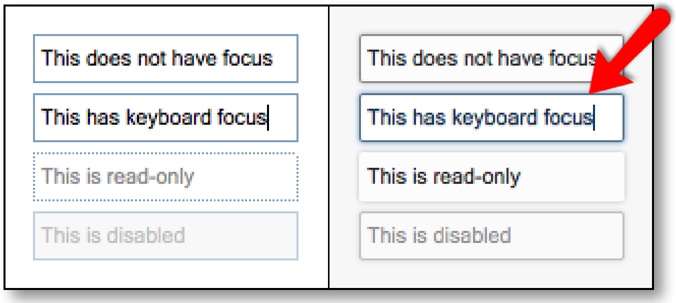
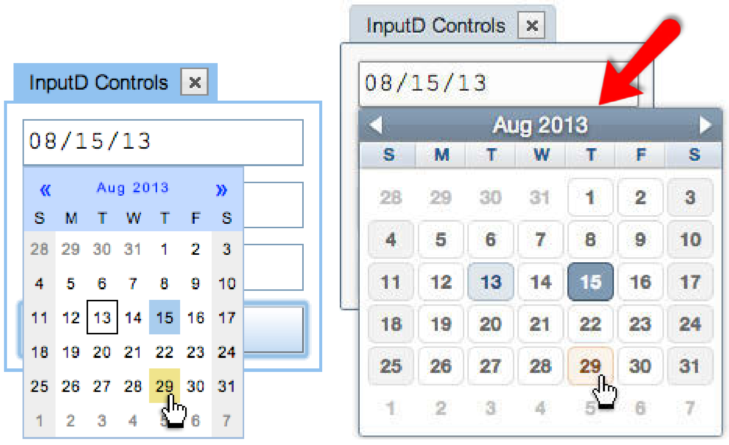


Figure 1. Comparing the old (left) and new (right) text controls



Aug 2013

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Aug 2013

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

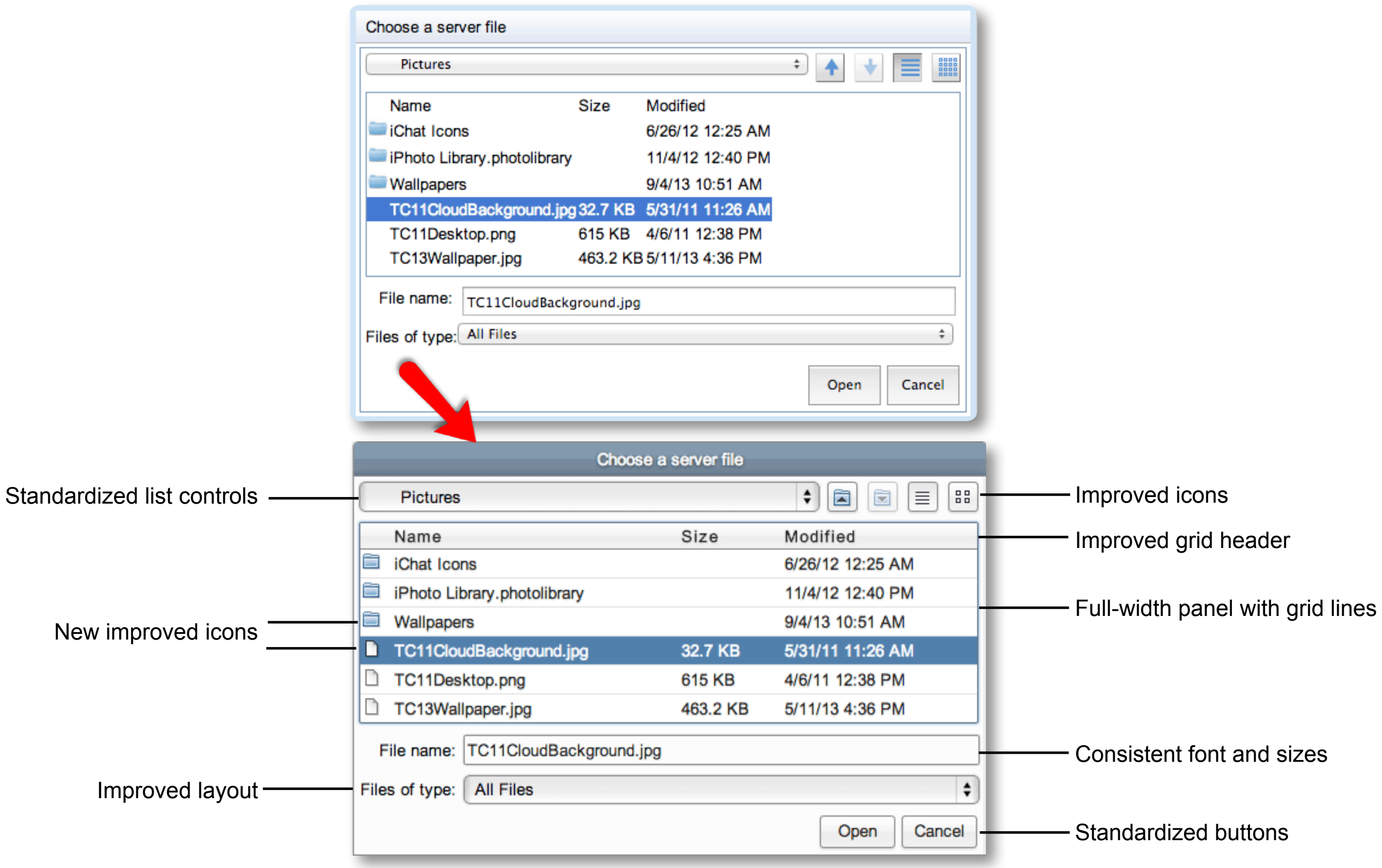
Figure 2. Comparing the old (left) and new (right) BBjInputD calendar

Improved Accessibility

In addition to providing better visual feedback to the user for various control states, the new theme also improves usability by increasing the size of interactive components. For example, MSGBOX buttons are larger and more consistently sized regardless of text, making it easier than ever to select the appropriate action. Spinner controls have improved horizontal and vertical layouts and their buttons are easier to click. The BBjInputD calendar (shown in **Figure 2**) is now larger, so navigating through months, selecting dates, and differentiating between weekends, weekdays, today's date, and the selected date is much more straightforward and less error-prone.

Improved Layouts

Although the CSS is responsible for much of the new theme, various other tweaks and changes polish the layout and appearance of many other language elements and constructs. A good example of this is the file chooser, shown in **Figure 3,** which is the basis for the FILEOPEN() and FILESAVE() functions and the BBjFileChooser controls. BASIS applied dozens of changes to improve the appearance of these choosers, including replacing the icons, improving the overall layout, standardizing the control types and sizes, improving the detail view header, and adding grid lines.



Name	Size	Modified
iChat Icons		6/26/12 12:25 AM
iPhoto Library.photolibrary		11/4/12 12:40 PM
Wallpapers		9/4/13 10:51 AM
TC11CloudBackground.jpg	32.7 KB	5/31/11 11:26 AM
TC11Desktop.png	615 KB	4/6/11 12:38 PM
TC13Wallpaper.jpg	463.2 KB	5/11/13 4:36 PM

File name: TC11CloudBackground.jpg
Files of type: All Files

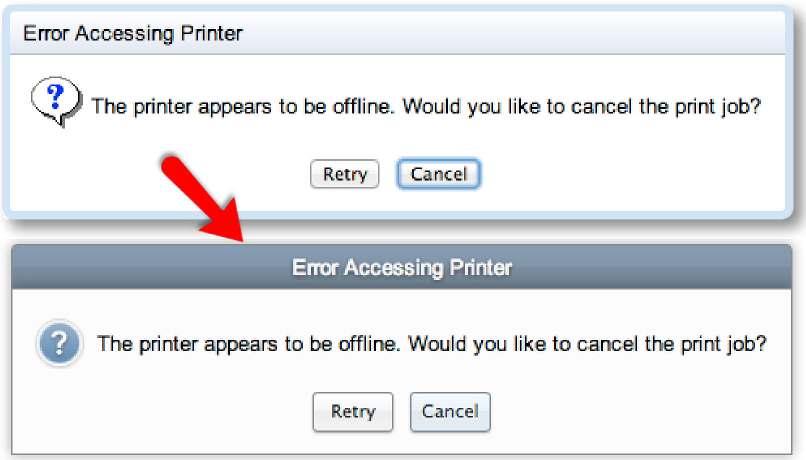
Name	Size	Modified
iChat Icons		6/26/12 12:25 AM
iPhoto Library.photolibrary		11/4/12 12:40 PM
Wallpapers		9/4/13 10:51 AM
TC11CloudBackground.jpg	32.7 KB	5/31/11 11:26 AM
TC11Desktop.png	615 KB	4/6/11 12:38 PM
TC13Wallpaper.jpg	463.2 KB	5/11/13 4:36 PM

File name: TC11CloudBackground.jpg
Files of type: All Files

Figure 3. Comparing the old (top) and the new (bottom) file chooser

Improved Controls

Because BBj is a fully featured language that offers dozens of user interface controls, the new theme had to cover a lot of ground. Some of the more basic controls, such as a BBjEditBox, have not changed radically. However, they now offer enhanced hover, focus, and read-only indicators. Complex componentized controls like the BBjGrid, BBjTabCtrl, and BBjInputD calendar practically begged for modification and so BASIS improved their appearance dramatically. BASIS also improved many aspects of an application that may not immediately come to mind – message boxes shown in **Figure 4**, authentication dialogs, the mini console – and even added a translucent glass pane that distinguishes the MSGBOX(), FILEOPEN(), and FILESAVE() functions from the underlying application to clearly indicate their modal nature.


Error Accessing Printer

The printer appears to be offline. Would you like to cancel the print job?

Error Accessing Printer

The printer appears to be offline. Would you like to cancel the print job?

Figure 4. Comparing the old (top) and the new (bottom) message boxes

Review

It is worth reiterating that any new or existing BUI app will benefit from the updated default theme without requiring any extra work on the developer's part. Simply update to BBj 14.0 and the new theme is now standard. BASIS put a lot of time and effort into designing the new theme, making almost 1,000 modifications to almost 800 selectors and augmenting the BUI-CSS interaction by adding scads of new selectors. This considerable endeavor results in your BUI applications looking and running better than ever, while also laying the groundwork for increased customization and control when creating your own theme with a CSS file that you provide. Update to 13.10 to preview BBj 14.0 features and instantly improve your BUI apps! ■

For more information, refer to
- *A Bountiful Selection of New CSS Selectors* online at links.basis.com/13selectors
- *Adding Style to BBx Web Apps With Custom CSS* at links.basis.com/11css
- *BBj BUI: Getting Started* at links.basis.com/buiguide
- *BUI CSS Component Styles* at links.basis.com/buicss

BARISTA®
Rapid
Application
Development
BARISTA® the ultimate RAD tool for database-driven software projects built with BBj®, the dynamic object-oriented programming language for Java
BASIS International
www.basis.com

Bisco Gets More Bang for Their BBx Buck

Like so many companies who have been a longtime BASIS customer, Bisco Industries (www.biscoind.com) found themselves wanting, "needing", to modernize the capabilities of their highly functional but dated-looking 20+ year old application. Bisco's BBx® application provided the utility necessary to meet their basic needs, but the application no longer met their growing desire for newer features and functions. Bisco staff members wanted a more modern graphical interface that would be easier to navigate with either a keyboard or a mouse, and better data access via reporting tools. Management was concerned about security and the potential risk of depending on a single server system.

When Brian Stover took over as IT Director, Bisco tasked him to either find another solution or make the existing application meet the new requirements. Brian quickly met the challenge to learn about the newest BBx generations. Adding to what he gleaned from past BASIS roadshows, he tapped into a variety of resources that included the Java Break video library (links.basis.com/javabreak), presentations from TechCon sessions, and BASIS personnel to create a technology roadmap for Bisco.

The Plan and Execution

After weighing the options – whether to move forward with BASIS technology or to a new system – Stover realized it was simply a matter of logic. Bisco could achieve their goals with BASIS' expanding technology better and faster than introducing an outside solution and without the concomitant risks to the business. More importantly, the BASIS solution was more cost effective, required less development, and came with an easier learning curve.

BASIS' flexibility allowed Bisco to transform the application in a carefully staged approach that had budgetary, scheduling, and risk-reduction benefits. This also enabled Bisco to immediately begin to deliver on the goal of bringing greater productivity to the workforce.

Bisco began working with BASIS Professional Services to establish some initial goals for transitioning from PRO/5® to BBj® and beginning to develop with Barista®. To meet those goals, they defined three steps; **1)** upgrade to current BASIS BBj technology, **2)** migrate to new three-tiered architecture to replace the 10-year old single-tier system, and **3)** explore Barista usage in a pilot project to replace the old CUI lookup system with a more efficient GUI lookup system.

Upgrade to BBj

By converting the old PRO/5 system to BBj, Bisco was able to move to three-tiered architecture. In addition to the scalability and reliability advantages of three-tier architecture, introducing BBj thin client with Java Web Start allowed Bisco to eliminate the additional cost of a third party terminal emulation product and allowed their users to run concurrent sessions without needing to increase the user count of their existing license. Using BBj's zero deployment capabilities also allowed for quicker and easier deployment of new users, which is critical for this growing business. BBj provided Bisco with access to BASIS' extended Structured Query Language (SQL) support, GUI form design capability, and support for AES 256-bit security. Should Bisco decide they want to deploy new mobile GUI applications in a browser, BBj's browser user interface (BUI) will also make this possible.

Migrate to Three-Tiered Architecture

At that time, Bisco Industries' application and data server consisted of a single five-year old T5220 SUN Solaris server. While this system was sufficient to run their application, there was no onsite redundancy and any type of server issue could bring this multi-million dollar company to a temporary standstill.

With guidance and assistance from the BASIS Professional Services team, Bisco implemented three-tier architecture consisting of four Dell PowerEdge R620 servers with multi-core 2.70 GHz processors and solid state drives. One server now acts as the data server with the other three configured as load balanced application servers. The desktop users access the application with BBj's thin client, deployed over their intranet via Java Web Start. Stover explains, "*I have found BASIS support to be number one. Our move from a single Solaris box to a three-tier Redhat solution, the conversion from PRO/5 to BBj, and development with Barista would have easily taken twice as long without the Professional Services from BASIS. BASIS continues to help us avoid the pitfalls that are associated with such a move. We have found that we can always get our question answered in a timely manner even when working with the European group. I don't think those guys ever sleep.*"

Laurence Guiney
Senior Account Manager

Bisco Industries is a premier distributor of electronic components and fasteners used for production in aerospace, communication, computer, fabrication, industrial equipment, instrumentation, marine, and military industries. Their mission is to be their customers' sole supplier of electronic components and fasteners by being a local presence, providing unsurpassed customer service and one-stop-shopping.

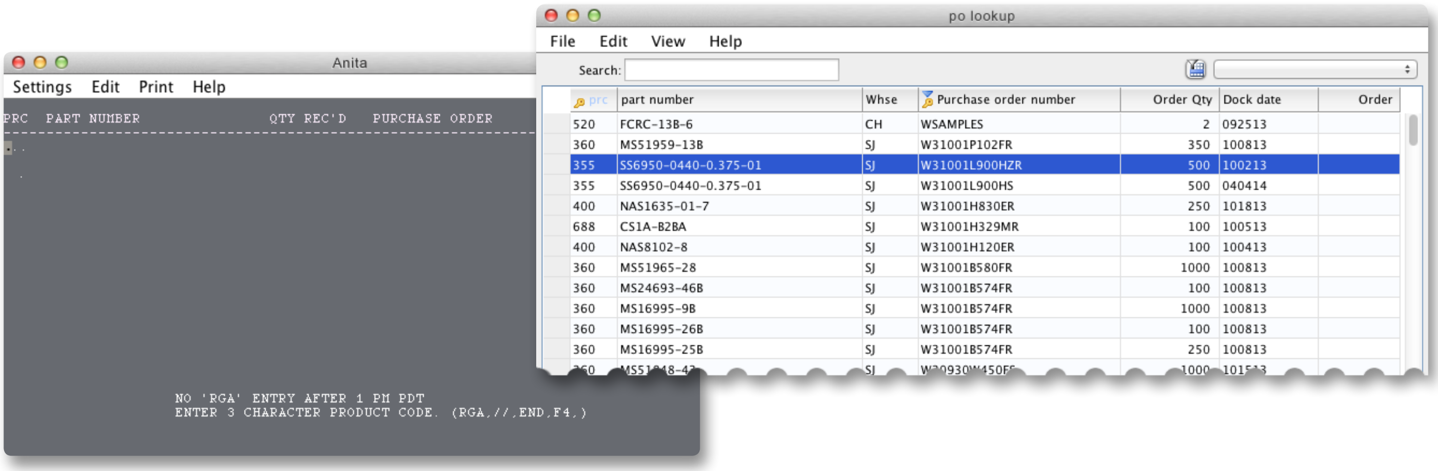
Bisco employs nearly 400 employees at 45 locations strategically located throughout the United States, Canada, and Mexico to ensure fast delivery and availability to their customers. Bisco's global expansion plans include entering the European market. www.biscoind.com

Three-tier architecture provides Bisco with increased flexibility, maintainability, reusability, and scalability. For high-availability purposes, Bisco will be adding a second database server soon to complete their high availability cluster and add additional robustness to their enterprise.

"...frees up one person to get the daily receiving process done..."

Transition Using Barista...With Sprinkles

To make Bisco's transition from CUI to GUI faster and more efficient, they chose to use Barista for their GUI development. The first phase replaced an old character-based lookup system with a new graphical lookup as shown in **Figure 1**.

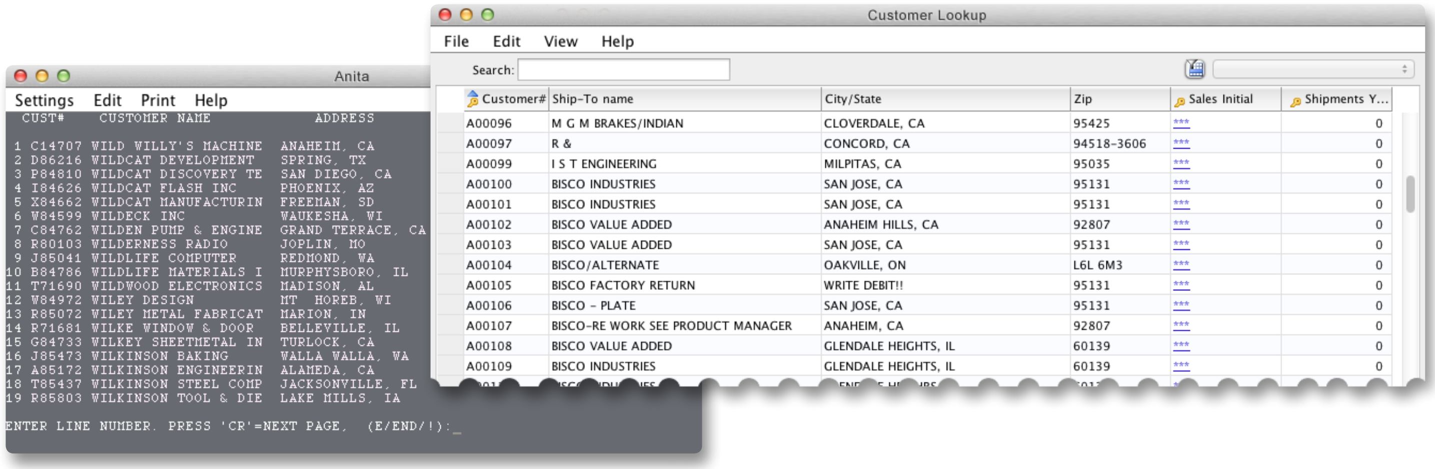


```
PRC  PART NUMBER                 QTY REC'D   PURCHASE ORDER
-------------------------------------------------------------
.. .

               NO 'RGA' ENTRY AFTER 1 PM PDT
               ENTER 3 CHARACTER PRODUCT CODE.  (RGA,//,END,F4,)
```

prc	part number	Whse	Purchase order number	Order Qty	Dock date	Order
520	FCRC-13B-6	CH	WSAMPLES	2	092513	
360	MS51959-13B	SJ	W31001P102FR	350	100813	
355	SS6950-0440-0.375-01	SJ	W31001L900HZR	500	100213	
355	SS6950-0440-0.375-01	SJ	W31001L900HS	500	040414	
400	NAS1635-01-7	SJ	W31001H830ER	250	101813	
688	CS1A-B2BA	SJ	W31001H329MR	100	100513	
400	NAS8102-8	SJ	W31001H120ER	100	100413	
360	MS51965-28	SJ	W31001B580FR	1000	100813	
360	MS24693-46B	SJ	W31001B574FR	100	100813	
360	MS16995-9B	SJ	W31001B574FR	1000	100813	
360	MS16995-26B	SJ	W31001B574FR	100	100813	
360	MS16995-25B	SJ	W31001B574FR	250	100813	

Figure 1. Comparing the Receiving look-up in the old CUI with the new GUI version

Bisco's new Barista version of the generalized lookup system – the Z Menu – provides their users a graphical lookup system to sort, filter, and apply wildcard searches to lookup customer or buyer information. **Figure 2** shows the CUI and new GUI version of the Z Menu lookups.



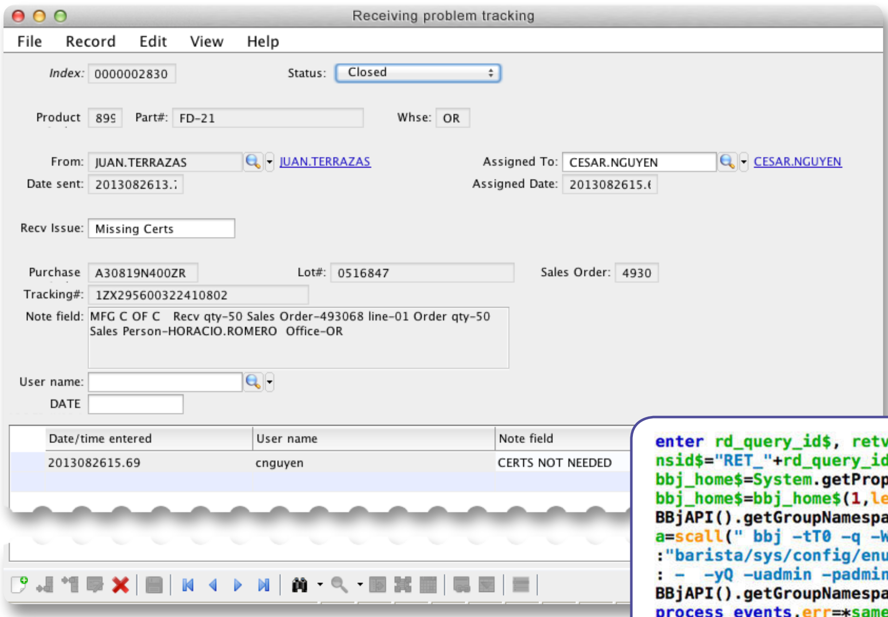
```
 CUST#    CUSTOMER NAME             ADDRESS

 1 C14707 WILD WILLY'S MACHINE  ANAHEIM, CA
 2 D86216 WILDCAT DEVELOPMENT   SPRING, TX
 3 P84810 WILDCAT DISCOVERY TE  SAN DIEGO, CA
 4 I84626 WILDCAT FLASH INC     PHOENIX, AZ
 5 X84662 WILDCAT MANUFACTURIN  FREEMAN, SD
 6 W84599 WILDECK INC           WAUKESHA, WI
 7 C84762 WILDEN PUMP & ENGINE  GRAND TERRACE, CA
 8 R80103 WILDERNESS RADIO      JOPLIN, MO
 9 J85041 WILDLIFE COMPUTER     REDMOND, WA
10 B84786 WILDLIFE MATERIALS I  MURPHYSBORO, IL
11 T71690 WILDWOOD ELECTRONICS  MADISON, AL
12 W84972 WILEY DESIGN          MT  HOREB, WI
13 R85072 WILEY METAL FABRICAT  MARION, IN
14 R71681 WILKE WINDOW & DOOR   BELLEVILLE, IL
15 G84733 WILKEY SHEETMETAL IN  TURLOCK, CA
16 J85473 WILKINSON BAKING      WALLA WALLA, WA
17 A85172 WILKINSON ENGINEERIN  ALAMEDA, CA
18 T85437 WILKINSON STEEL COMP  JACKSONVILLE, FL
19 R85803 WILKINSON TOOL & DIE  LAKE MILLS, IA

ENTER LINE NUMBER. PRESS 'CR'=NEXT PAGE,  (E/END/!):_
```

Customer#	Ship-To name	City/State	Zip	Sales Initial	Shipments Y...
A00096	M G M BRAKES/INDIAN	CLOVERDALE, CA	95425	***	0
A00097	R &	CONCORD, CA	94518-3606	***	0
A00099	I S T ENGINEERING	MILPITAS, CA	95035	***	0
A00100	BISCO INDUSTRIES	SAN JOSE, CA	95131	***	0
A00101	BISCO INDUSTRIES	SAN JOSE, CA	95131	***	0
A00102	BISCO VALUE ADDED	ANAHEIM HILLS, CA	92807	***	0
A00103	BISCO VALUE ADDED	SAN JOSE, CA	95131	***	0
A00104	BISCO/ALTERNATE	OAKVILLE, ON	L6L 6M3	***	0
A00105	BISCO FACTORY RETURN	WRITE DEBIT!!	95131	***	0
A00106	BISCO - PLATE	SAN JOSE, CA	95131	***	0
A00107	BISCO-RE WORK SEE PRODUCT MANAGER	ANAHEIM, CA	92807	***	0
A00108	BISCO VALUE ADDED	GLENDALE HEIGHTS, IL	60139	***	0
A00109	BISCO INDUSTRIES	GLENDALE HEIGHTS, IL	60139	***	0

Figure 2. Comparing the old Z-Menu lookup screen in CUI with the new version in GUI

According to Mark Wright, lead developer on Bisco's migration to GUI, *"The new receiving lookup has made the function of receiving in the warehouse much quicker. The warehouse staff is able to lookup the 'Purchase order number' by PRC/Part#, Vendor#, or warehouse just by clicking on the field they want to search and even use a wildcard '%'. It has become a standard feature all departments are now looking to have."* Wright continues, *"The old receiving screen would require the users to use four screens to have the same function as the one Barista grid lookup. The time savings for just one of our larger warehouses frees up one person to get the daily receiving process done, down from four people to three."*

Wright goes on to describe the completely new functionality that Barista made possible with its easy-to-use rapid development features, "*The development could have been done in BBj with some work adding more keys and a lot of code. Instead, we were able to complete the Barista lookup in less than a hour and a few lines of code in the CUI program to make it all happen.*" Wright adds, *"We also created a new process to track receiving problems completely from Barista. This look-up functionality took me less than a day from start to finish; that includes the look-ups and filters for each department. There is not much in training for users since all the Barista screens function the same way.*"



Index: 0000002830 Status: Closed

Product 899 Part#: FD-21 Whse: OR

From: JUAN.TERRAZAS JUAN.TERRAZAS Assigned To: CESAR.NGUYEN CESAR.NGUYEN
Date sent: 2013082613.1 Assigned Date: 2013082615.6

Recv Issue: Missing Certs

Purchase A30819N400ZR Lot#: 0516847 Sales Order: 4930
Tracking#: 1ZX295600322410802
Note field: MFG C OF C Recv qty-50 Sales Order-493068 line-01 Order qty-50 Sales Person-HORACIO.ROMERO Office-OR

User name:
DATE

Date/time entered	User name	Note field
2013082615.69	cnguyen	CERTS NOT NEEDED

Figure 3. Barista-developed receiving tracker

"...less than a hour and a few lines of code..."

```
enter rd_query_id$, retval$
nsid$="RET_"+rd_query_id$+str(java.util.UUID.randomUUID())
bbj_home$=System.getProperty("com.basis.server.configDirectory")
bbj_home$=bbj_home$(1,len(bbj_home$)-3)
BBjAPI().getGroupNamespace().setValue(nsid$,"")
a=scall(" bbj -tT0 -q -WD"""+bbj_home$+"barista/""  -c"""+bbj_home$+
:"barista/sys/config/enu/barista.cfg""  sys/prog/bax_launch_task.bbj
: -  -yQ -uadmin -padmin123 -q"+rd_query_id$+" -mINQ  -n"+nsid$+"  ")
BBjAPI().getGroupNamespace().setCallbackForVariableChange(nsid$,"retval")
process_events,err=*same
retval:
BBjAPI().getGroupNamespace().removeCallbackForVariableChange(nsid$)
retval$ = BBjAPI().getGroupNamespace().getValue(nsid$)
if retval$=$00$ or retval$="::CANCEL::" then let retval$=""
exit
```

Figure 4. Actual code that added additional functionality to Bisco's application

Figure 3 shows a sample result from this new tracking process.

By using Barista and its "sprinkles," Bisco was able to take advantage of the automatically generated grids. Sprinkles include utilization of the GUI look-up grid and built in filtering, re-ordering, print preview, and support of numerous output file types including PDF, XLS, CSV, XML, and Google Docs, as well as the ability to fax, email, and archive documents. Bisco added these sprinkles with just a few lines of the code (**Figure 4**) in their CUI app.

Another Need Met

In addition to the initial issues Bisco addressed with BASIS, their webstore didn't quite interface to their ERP solution as well as they wanted. Learning about the AddonStore™, a webstore based on BBj technology, Bisco decided to implement the AddonStore as their new online interface to their existing ERP solution. This next project will not only provide them with the means to easily update the look of their site and allow for further expansion at a later date, but also give them a number of other benefits:

- The AddonStore provides easier development with separation between design and code, and a modern object-oriented development paradigm.
- The AddonStore uses the same technology for the internal system as Bisco's internal system, which allows for the re-use of business logic as well as the data formats (the store will receive the data using BBj replication).
- The AddonStore also allows more extensive use of cascading style sheets (CSS) than the current Bisco site, enabling them to apply different styles to the original content to give it a completely different look and easily change it at any time to keep the look and feel fresh and modern.

Bisco is again looking to BASIS Professional Services for assistance as they adapt and customize the AddonStore, and train Bisco employees to take over maintenance and modification of the site. Bisco programmers are learning by working on the very same site that they will eventually be responsible for, instead of simply learning from traditional training material.

Summary

Since facing this decision a few years ago to update their business application, Bisco has successfully provided their employees with the beginnings of a modern GUI application that combines the functionality of their existing business solution with the ease-of-use and look of a modern graphical user interface. By choosing to upgrade to BBj with integral assistance from BASIS Professional Services, Bisco is now transforming both their in-house application as well as their website to be functional, attractive, and user friendly. Bisco now has a robust enterprise with redundancy and performance, and room for unlimited expansion by adding additional servers to the load balanced cluster. They continue to add GUI functionality with Barista and with the AddonStore, and will look for even more ways to take advantage of BASIS technology in the future.

BASIS Professional Services played a key role in assisting with Bisco's survey of how well BASIS technology could fulfill their new requirements and by training Bisco's IT staff to continue meeting the needs and requirements of customers and employees alike, well into the future. If you find yourself in the same place as Bisco was, wanting or "needing" to modernize the look and capabilities of your functional but dated-looking application, consider sharing the task with BASIS Professional Services as Bisco did, and look forward to enjoying a fresher, more efficient, more modern you. ■ links.basis.com/**13code**

A Bountiful Selection of New CSS Selectors

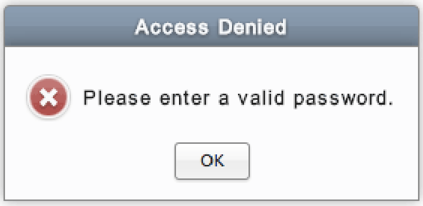
While the BBj® browser user interface (BUI) has taken advantage of cascading style sheets (CSS) ever since its introduction, BASIS has shifted its CSS capabilities into overdrive with the release of BBj 14.0! One of the most noticeable changes is that all BUI apps benefit from a brand new default look and feel. The underlying changes that made the new theme possible were due to the addition of several new CSS selectors to most of the BBj controls. This article gives an overview of how the new selectors allowed BASIS to update BUI in general, and how they give developers the power to create even more advanced and customized user experiences.

Selectors for Complex Controls

In the same way website designers use CSS selectors to apply styling to specific areas of a web page, BBj developers can use them to identify controls that make up the graphical user interface of their application. Because CSS selectors are flexible, you can use them to select entire classes of controls (e.g. all BBjButtons), single controls (e.g. the [Exit] button), or even portions of complex controls (e.g. the [OK] button on a message box). The message box is a prime example of one of BBj's more complex controls as it is composed of various pieces including the

- Dialog window
- Title bar
- Message icon
- Message text
- Response buttons (e.g. Yes/No/Cancel)

All of the components are combined to provide the user with a functional message box, an example of which is shown in **Figure 1**.


Access Denied

Please enter a valid password.

OK

Figure 1. A sample message box

Figure 2 shows the same message box, but this time BASIS added custom CSS to give a 2-pixel colored border to the various components, thus making it easier to visualize the discrete pieces that make up a standard message box.

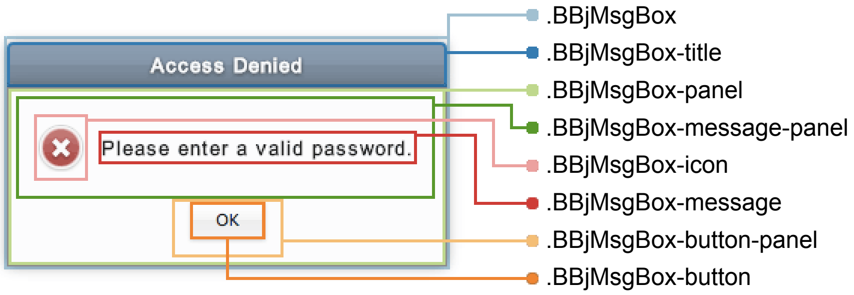


Figure 2. A sample message box identifying each of its selectors

By Nick Decker
Engineering Supervisor

Why Add More Selectors?

In order to customize the appearance effectively of complex controls like the message box, developers need to be able to specify how the individual pieces of the control should appear. This is somewhat analogous to painting an automobile, as you'd never dream of painting your entire car the same color. Various parts of the car are made up of different materials and colors, such as chrome siding, black tires, and white pin striping, all of which work in concert to provide detail, interest, and contrast. Without that same level of influence over a control's components, it would be impossible to change the appearance of a complex control like the message box effectively. This is precisely the reason BASIS has added numerous new selectors to a variety of BBj controls – to give developers the power and flexibility to modify the look of their application with a fine level of granularity.

Applying the Message Box Selectors

In addition to adding the new selectors, BASIS created extensive documentation covering all of the BUI CSS Component Styles. Find the Message Box selectors listed under BBjMsgBox, which shows a dozen different selectors that provide developers with the power to modify the style of each of the message box's components. Developers can now create custom definitions for several of the available message box selectors, transforming it into something that better matches the style of the BASIS web app as shown in **Figure 3**.

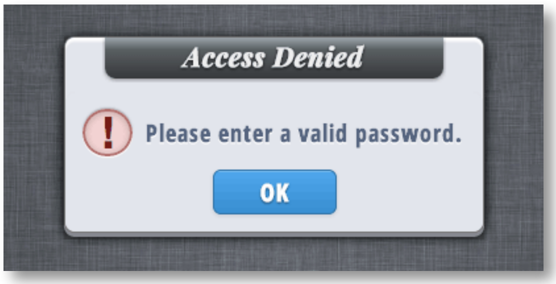


Figure 3. The same message box with custom CSS applied

This sample demonstrates how BASIS dramatically changed the appearance of the message box's components by

- modifying the border and adding a drop shadow to the dialog,
- changing the background color of the dialog to match the app,
- modifying the size, font, color, and hover/active states of the OK button,
- replacing the error icon image with infinitely scalable pure CSS,
- modifying the color, gradient, drop shadow, font, and side padding of the title bar, and
- changing the font and color, and adding an embossed effect to the message.

To accomplish all of these changes, the developers used a custom CSS file that modified the appearance of the message box's components via the selectors shown previously in **Figure 2**. Notice that the customizations changed the icon from the default image showing a red 'X' to a more subtle '!' that may look a little less ominous to users.

While you may not choose to go to this level of customization for your application, this sample serves as a great illustration of the extent to which you can take your BUI customizations. Not only did the developers replace the default image, but they also created its replacement out of pure CSS. Because this technique does not require any images, the new error 'icon' is infinitely scalable and will look clear and sharp – even if the user repeatedly enlarges the BUI app with the browser's zoom functionality. The CSS that achieved this customization appears in **Figure 4**.

```
.customMsgBox .BBjMsgBox-icon-error {

    /*  Since we're going to do something different with the icon, we have to set the
        size to ensure it will show up, then we can customize it.                    */
        border-radius: 40px;
        background: hsl(0,80%,90%);
        width: 34px;
        height: 32px;
        padding: 0;
        border: 2px solid rgba(128,0,0,.5);
        box-shadow: inset 0px 1px 8px rgba(0,0,0,0.2), 0 1px 1px #fff, 0 -1px 1px rgba(0,0,0,0.2);
}

.customMsgBox .BBjMsgBox-icon-error img {

    /*  This tells the original icon not to show, since we will
        be replacing the image with a pure-css indicator        */
        display: none;
}

.customMsgBox .BBjMsgBox-icon-error:after {

    /*  The :after means that we'll be adding this information to the selector.
        We start by adding content (!), then customizing its appearance          */
        content: '!';
        text-align: center;
        float: left;
        width: 100%;
        color: rgb(128,0,0);
        font: bold 2.5em/1.0em "Helvetica Neue", Arial, Helvetica, Geneva, sans-serif !important;
}
```

Figure 4. Custom CSS that replaces the error icon

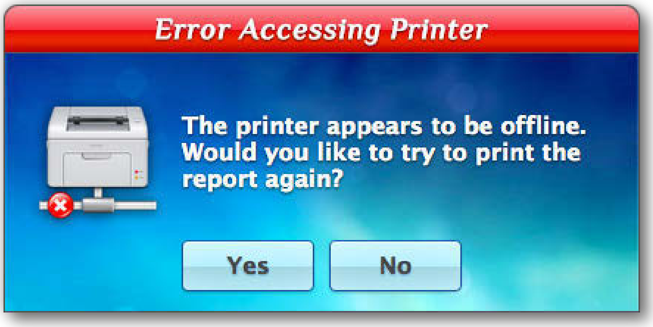
The custom CSS replaces the default error icon in the three steps listed below, using the *.BBjMsgBox-icon-error* selector, thus specifically targeting the error icon and leaving the question, warning, and info icons intact.

Step 1 – Modifies the look of the HTML div that normally displays the error icon. This does it by specifying values for the border-radius to round the corners and turn it into a circle, changing the background color to a light red, setting the size via the width and height properties, adding a translucent dark red border, and adding some 3D effects via the box-shadow property.

Step 2 – Ensures that those changes are visible to the user now that the div is styled. This does it by hiding the red 'X' image icon that normally displays, revealing the underlying div with the customizations.

Step 3 – Adds the exclamation mark to the div to complete the replacement. The CSS accomplishes this by taking advantage of the :after pseudo-element, which appends new content to the div's original content. In this example, the div was only used as a container for the icon image, but now that the image has been hidden in Step 2, you are free to add new content to the div. In addition to adding the exclamation mark, this step also defines how the new content will appear. It defines the font, color, and placement of the exclamation mark so that it is perfectly centered in the div.

While the previous example replaced the error icon with pure CSS, it is certainly possible to replace the icon with an image of your own. In fact, you can use custom images for other selectors as well. The screenshot in **Figure 5** shows yet another incarnation of the BBj message box with custom CSS that does exactly that. This time the CSS applied a bold, colorful theme to the entire control and made use of images to customize the body of the message box as well providing a fancy replacement for the error icon.



Error Accessing Printer

The printer appears to be offline. Would you like to try to print the report again?

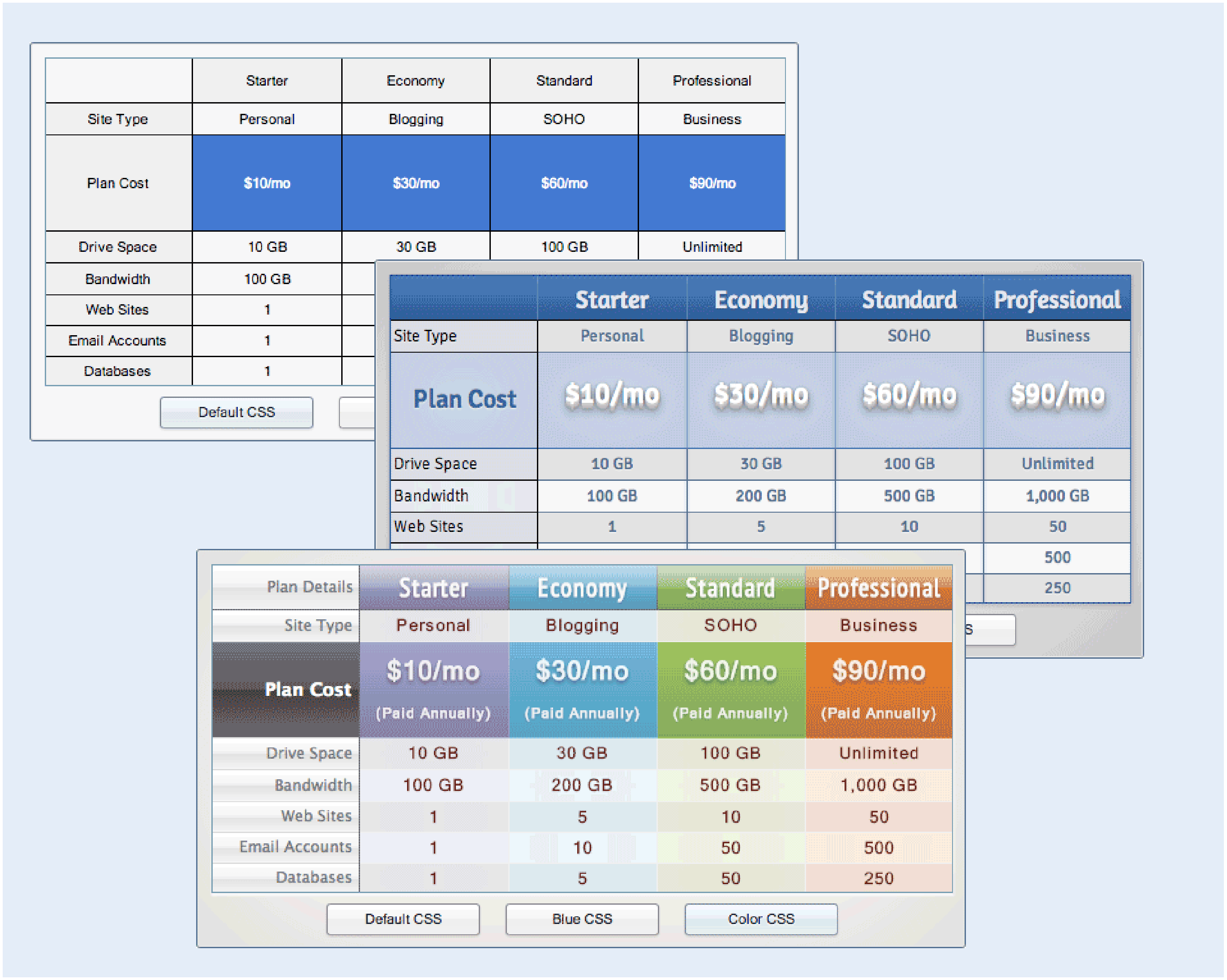
Figure 5. CSS using images to personalize the icon and background of the message box

Utilizing the Grid's New Selectors

The BBjGrid serves as the last example, and boy does it offer a fantastic opportunity for customization! BASIS added over 30 new selectors to the grid control alone, which is especially impressive once you realize that many of these selectors will often be used in conjunction with one another to offer countless combinations.

By way of example, developers can change the color of the grid's selected cell or row and still maintain alternating row colors by using the `.BBjGrid-evenRow` and `.BBj-selected` style names together to target the selected even-numbered row.

Figure 6 shows screenshots of the live GridCSS demo, which uses a standard BBjGrid to display a pricing comparison chart for a WebSite Hosting plan. It offers three distinct representations of the grid via custom CSS: the BBj default, a modified blue theme, and a multicolor theme. BASIS made good use of the new selectors to modify various aspects of the grid, including fonts, colors, headers, selected row, and border color.



	Starter	Economy	Standard	Professional
Site Type	Personal	Blogging	SOHO	Business
Plan Cost	$10/mo	$30/mo	$60/mo	$90/mo
Drive Space	10 GB	30 GB	100 GB	Unlimited
Bandwidth	100 GB			
Web Sites	1			
Email Accounts	1			
Databases	1			

	Starter	Economy	Standard	Professional
Site Type	Personal	Blogging	SOHO	Business
Plan Cost	$10/mo	$30/mo	$60/mo	$90/mo
Drive Space	10 GB	30 GB	100 GB	Unlimited
Bandwidth	100 GB	200 GB	500 GB	1,000 GB
Web Sites	1	5	10	50
				500
				250

Plan Details	Starter	Economy	Standard	Professional
Site Type	Personal	Blogging	SOHO	Business
Plan Cost	$10/mo (Paid Annually)	$30/mo (Paid Annually)	$60/mo (Paid Annually)	$90/mo (Paid Annually)
Drive Space	10 GB	30 GB	100 GB	Unlimited
Bandwidth	100 GB	200 GB	500 GB	1,000 GB
Web Sites	1	5	10	50
Email Accounts	1	10	50	500
Databases	1	5	50	250

Figure 6. The GridCSS demo showing the default version along with two custom themes

More Componentized Controls

The message box and grid are just a couple of examples of a componentized control. Many others exist, such as the BBjTabCtrl, BBjMenuButton, BBjNavigator, and so on. These complex controls now have several selectors available, giving you the ability to access and change the many pieces that make up the complete control. The more selectors a particular control offers, the more potential you have to define how the element appears and acts.

Summary

BBj controls now offer several hundred selectors, allowing a never-before-seen level of customization over all of the popular graphical controls. Combining those selectors with one another or the available control states such as read-only, disabled, bordered, focused, default, selected, checked, etc. result in countless combinations and infinite customizability. Remember to check out the BUI CSS Component Styles document to learn about the new selectors. See the sample CSS and how it modifies each control at a low level, and then run the linked demos. CSS is ready and waiting for you. Go forth and customize! ■

links.basis.com/**13code**

For more information, refer to
- *BUI CSS Component Styles* at links.basis.com/buicss
- *Adding Style to BBx Web Apps With Custom CSS* at links.basis.com/11css

Barista Creations – Add Your App, Stir, and Enjoy

One of the strongest incentives for developing within the Barista® Application Framework is the instant availability of new Barista features and functionality to your existing applications. Read on to learn how you can provide configurable, more sophisticated error handling in your Barista applications, and how to unleash the power of filtering in Barista queries.

Console Access

Console access has long been one of the most amazing benefits of programming in BASIS' interpretive BBx® languages. This is especially true when dealing with unexpected errors. This powerful feature can be particularly handy when dealing with an application runtime framework like Barista. With these new error-handling enhancements, Barista delivers the troubleshooting control that developers have grown to expect from their BBx-based applications.

By Christine Hawkins
Software Developer

Handling the Unexpected

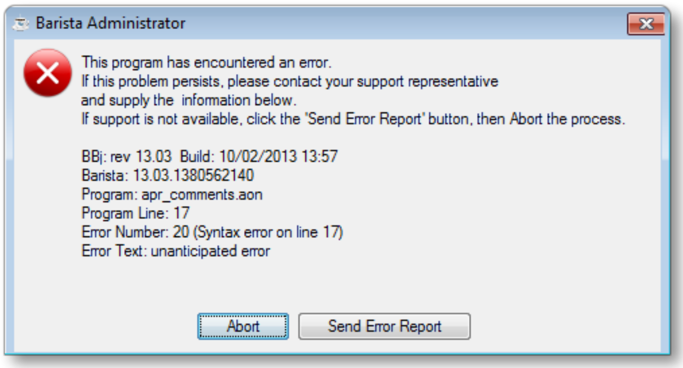
Barista's messaging system already provides a robust mechanism for interacting with users when an application encounters an anticipated error or processing anomaly. Now, the process for handling *unanticipated* errors is enhanced so that Barista application developers can configure and control access to the console as well as error reporting options.

Three levels of error handling are available so choose the level that suits your needs. Make your choice based on the type of system, e.g., production, demonstration, or development, and the desired level of access that should be granted to users who may encounter an unanticipated error.

Strict:	No console access is allowed.
Authorized:	Console access is permitted, but only if a password is supplied.
Open:	Console access is granted.

Strict Error Handling

Strict error handling does not permit any real-time debugging. Users may be able to retry/resume processing if the error is due to a locked record or file, but otherwise they must abort the process or send an error report. This is the mode that is in effect by default when launching applications via Web Start. **Figure 1** shows an example of what happens when an unanticipated error occurs in this mode.


Barista Administrator

This program has encountered an error.
If this problem persists, please contact your support representative
and supply the information below.
If support is not available, click the 'Send Error Report' button, then Abort the process.

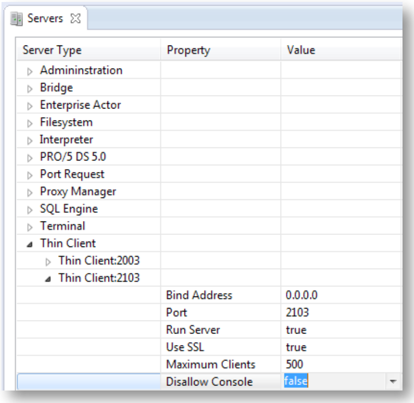
BBj: rev 13.03 Build: 10/02/2013 13:57
Barista: 13.03.1380562140
Program: apr_comments.aon
Program Line: 17
Error Number: 20 (Syntax error on line 17)
Error Text: unanticipated error

Figure 1. Unanticipated error message showing detailed information about the error

Authorized Error Handling

If you want to permit access to the console in a controlled fashion, configure Authorized error handling with two easy steps. First, remove the Disallow Console setting in BBj using Enterprise Manager (see **Figure 2**).

Next, configure Barista to display a custom message, and accept a password before allowing access, as shown in these lines from barista.cfg:


Servers

Server Type	Property	Value
Administration		
Bridge		
Enterprise Actor		
Filesystem		
Interpreter		
PRO/5 DS 5.0		
Port Request		
Proxy Manager		
SQL Engine		
Terminal		
Thin Client		
Thin Client:2003		
Thin Client:2103		
	Bind Address	0.0.0.0
	Port	2103
	Run Server	true
	Use SSL	true
	Maximum Clients	500
	Disallow Console	false

Figure 2. Turn off the Disallow Console setting in the Servers tab of the BBj Services node

```
set !CONEXIT=true
set !CONMESS=Please provide the password for console access, or <enter> to retry:
set !CONPASS=admin123
```

Authorized mode provides an additional [Debug] button on the error dialog so the console can be accessed once the user supplies the correct password.

Open Error Handling

If there are no concerns about console access, use Open mode. Like Authorized mode, Open mode removes the Disallow Console restriction in Enterprise Manager. However, with Open mode you do not enable the !CONxxx globals to prompt for an additional password, so clicking the [Debug] button immediately drops the user to a console prompt.

To help developers in the debugging process, BASIS unprotected the code in several of the Barista form-related runtime programs. This makes the debugging option available if an error occurs in one of these Barista programs when using Authorized or Open mode. Although the code is unprotected for debugging purposes, this is not an invitation to modify it; any modifications render the program unusable. Should a developer accidentally do so, the only recourse would be to restore the original program.

Figure 3 shows code from an AddonSoftware® program that uses the Barista error handler. In this sample, if the code isn't protected (as indicated by `tcb(2)=0)`, then the text from the error line is fed into the handler. Once in the handler, if console access is allowed and the error line text has been supplied, the [Debug] button will be included on the error message dialog. Barista returns "ESCAPE" or "RETRY" if the user has pushed the [Debug] or [Retry] buttons in the message dialog, so the program can take action accordingly, or do an exit or release if the user has opted to Abort.

```
std_error:  rem --- Standard error handler

    rd_err_text$=""
    if tcb(2)=0 and tcb(5) then rd_err_text$=pgm(tcb(5),tcb(13),err=*next)
    call stbl("+DIR_SYP")+"bac_error.bbj",pgm(-2),str(tcb(5)),str(err),rd_err_text$,rd_err_act$
    if pos("ESCAPE"=rd_err_act$) seterr 0; setesc 0
    if pos("RETRY"=rd_err_act$) retry
    if pgm(-1)<>pgm(-2) status=999; exit
    release
```

Figure 3. Sample code for calling the Barista error handler

Regardless of the error handling mode, users can always generate an error report and add them to Barista's Document Processing Queue to email to the designated recipient. The error report form provides text input so users can describe the run-time conditions surrounding the error, and also a checkbox for including a workspace memory dump as part of the error report. Developers or system administrators can configure Barista to allow or deny inclusion of workspace memory dumps, or include them if a password is supplied, as seen in **Figure 4.**

WHERE There's a Filter, There's a Way (to get the data you want)!

Inquiry

The inquiry system is perhaps Barista's most popular built-in component. Users can quickly and easily launch inquiries on forms and individual fields to find the data they're looking for, or click hyperlinks to display complete information for a coded field. In addition, applications can tie into the inquiry system with drilldowns and custom queries. The queries themselves are loaded with features for sorting, searching, filtering, and exporting in any of several formats (**Figure 5**).

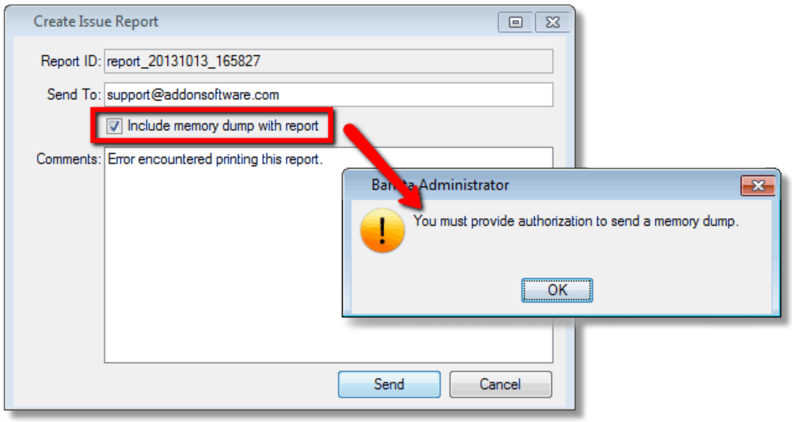
Filter

If the basic sorting and searching capabilities aren't enough, users can take it further with the built-in filtering tool. The enhanced point and click interface in the filter tool makes it easy to construct more complicated filters across multiple columns, using AND/OR conjunctions, various operators, and parentheses for grouping. Power users may even be allowed to access the WHERE clause for direct editing. And remember, any number of filters can be defined and saved, so a user may quickly recall a filter to run the query again later, rather than having to reconstruct it every time, delivering mini-reportwriter functionality to the user.

Figure 6 shows the filter tool in action. Toggle the filter tool on or off by clicking the filter button at the top right of the query form. Create a filter by selecting the desired column, operator, and value, along with the desired conjunction. Click the [Enter] button to add each filter component of the resulting WHERE clause to the box at the right. When the clause contains all of the desired filter components, press the [Execute] button to run the query using the clause and see the results.

Group

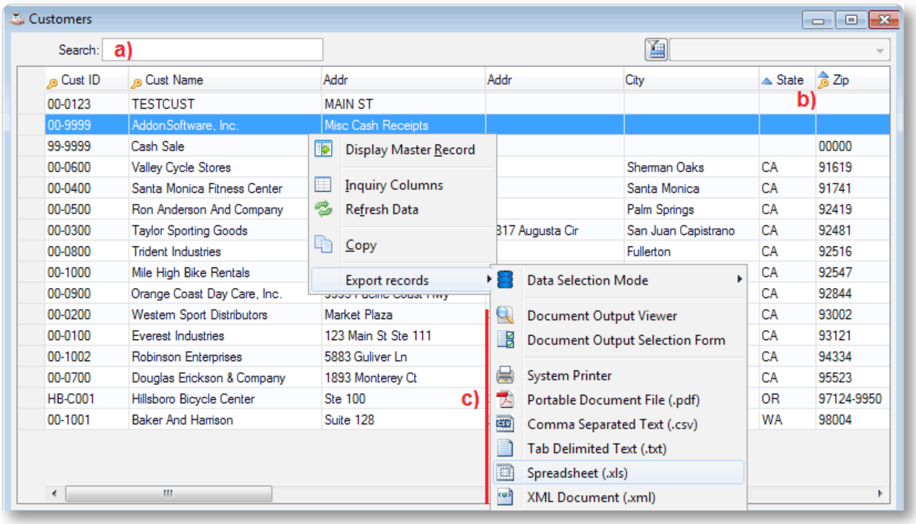
In **Figure 6**, the query results aren't quite as expected. Some grouping is



Report ID: report_20131013_165827
Send To: support@addonsoftware.com
[✓] Include memory dump with report
Comments: Error encountered printing this report.

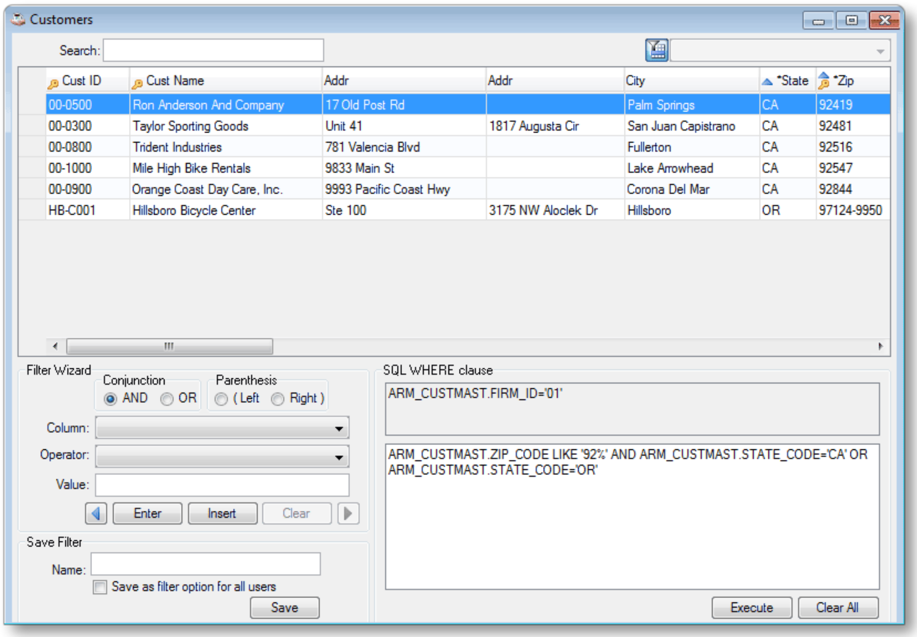
You must provide authorization to send a memory dump.

Figure 4. Create Issue Report and optionally allow inclusion of a workspace memory dump



Cust ID	Cust Name	Addr	Addr	City	State	Zip
00-0123	TESTCUST	MAIN ST				
00-9999	AddonSoftware, Inc.	Misc Cash Receipts				
99-9999	Cash Sale					00000
00-0600	Valley Cycle Stores			Sherman Oaks	CA	91619
00-0400	Santa Monica Fitness Center			Santa Monica	CA	91741
00-0500	Ron Anderson And Company			Palm Springs	CA	92419
00-0300	Taylor Sporting Goods		317 Augusta Cir	San Juan Capistrano	CA	92481
00-0800	Trident Industries			Fullerton	CA	92516
00-1000	Mile High Bike Rentals				CA	92547
00-0900	Orange Coast Day Care, Inc.				CA	92844
00-0200	Western Sport Distributors	Market Plaza			CA	93002
00-0100	Everest Industries	123 Main St Ste 111			CA	93121
00-1002	Robinson Enterprises	5883 Guliver Ln			CA	94334
00-0700	Douglas Erickson & Company	1893 Monterey Ct			CA	95523
HB-C001	Hillsboro Bicycle Center	Ste 100			OR	97124-9950
00-1001	Baker And Harrison	Suite 128			WA	98004

Figure 5. Inquiry grid showing a) Search box for quick filtering, b) multi-column sorting for State and Zip, and c) a variety of ways to output the query results



Cust ID	Cust Name	Addr	Addr	City	*State	*Zip
00-0500	Ron Anderson And Company	17 Old Post Rd		Palm Springs	CA	92419
00-0300	Taylor Sporting Goods	Unit 41	1817 Augusta Cir	San Juan Capistrano	CA	92481
00-0800	Trident Industries	781 Valencia Blvd		Fullerton	CA	92516
00-1000	Mile High Bike Rentals	9833 Main St		Lake Arrowhead	CA	92547
00-0900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy		Corona Del Mar	CA	92844
HB-C001	Hillsboro Bicycle Center	Ste 100	3175 NW Aloclek Dr	Hillsboro	OR	97124-9950

Conjunction: [✓ AND] OR
Parenthesis: (Left Right)
[ ] Save as filter option for all users

```
ARM_CUSTMAST.FIRM_ID='01'
```

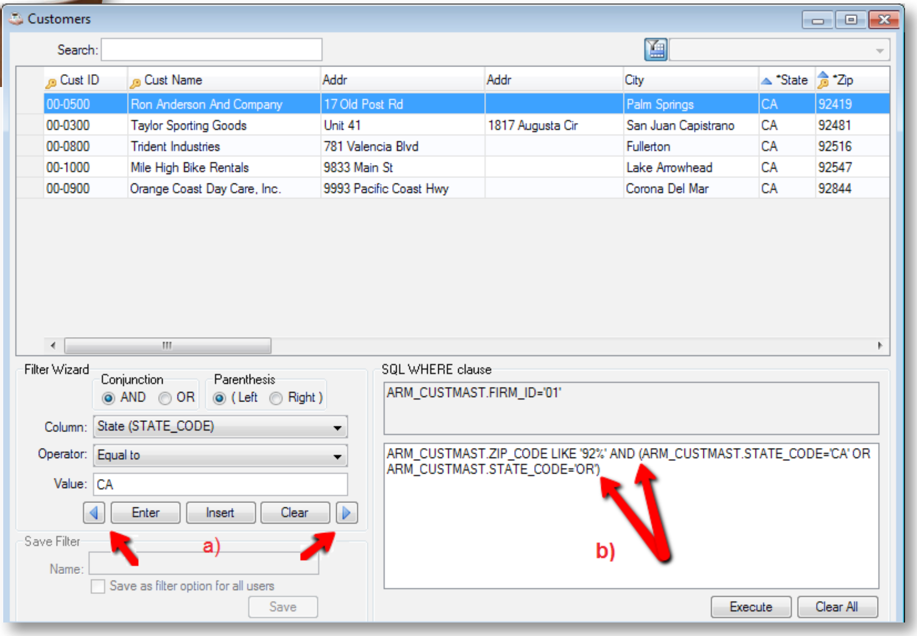
```
ARM_CUSTMAST.ZIP_CODE LIKE '92%' AND ARM_CUSTMAST.STATE_CODE='CA' OR
ARM_CUSTMAST.STATE_CODE='OR'
```

Figure 6. Filter for customer zip codes beginning with "92" and in the state of "CA" or "OR"

necessary to see to it that the value specified for the zip code applies across both states. Use the left and right arrow buttons (**Figure 7a**) to move back and forward through the WHERE clause one unit or filter component at a time. The wizard fields automatically populate with the column, operator, value, etc. of the filter component with the "focus." Then click the *Left* or *Right* radio buttons followed by the [Enter] button to add a parenthesis at the desired location (**Figure 7b**).

In addition to being able to specify the AND/OR conjunction and add parentheses, the point and click filter tool permits insertion or deletion of filter components. Use the arrow buttons in the Filter Wizard group box to "scroll" to the desired filter component, then press [Clear] to remove it from the clause, or [Insert] to create a new filter component in front of the one displayed.

All users can see the WHERE clause taking shape when working with the point and click interface, but Barista security options also make it possible for power users or administrators to edit the WHERE clause directly. An additional STATE_CODE filter component has been added directly to the clause in **Figure 8** by copying one of the other STATE_CODE filter components, then pasting and changing the state to "WA." Even with these permissions, the most efficient way to build a filter is usually a combination of point and click to do the main construction, and then direct edit for fine-tuning.



Customers

Search:

Cust ID	Cust Name	Addr	Addr	City	*State	*Zip
00-0500	Ron Anderson And Company	17 Old Post Rd		Palm Springs	CA	92419
00-0300	Taylor Sporting Goods	Unit 41	1817 Augusta Cir	San Juan Capistrano	CA	92481
00-0800	Trident Industries	781 Valencia Blvd		Fullerton	CA	92516
00-1000	Mile High Bike Rentals	9833 Main St		Lake Arrowhead	CA	92547
00-0900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy		Corona Del Mar	CA	92844

Filter Wizard

Conjunction: [✓ AND] OR

Parenthesis: [✓ (Left] Right)

Column: State (STATE_CODE)

Operator: Equal to

Value: CA

Save Filter

Name:

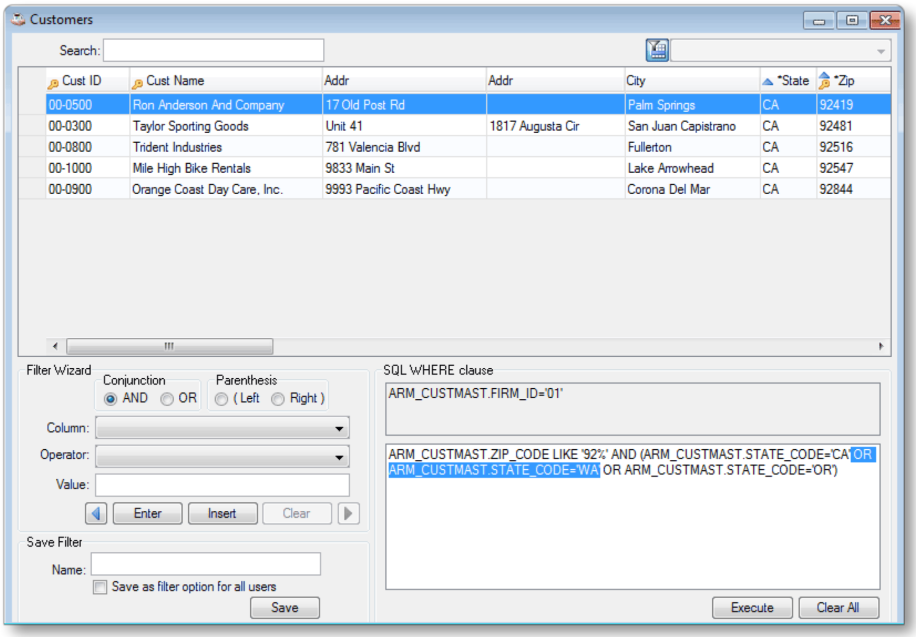
Save as filter option for all users

SQL WHERE clause

```
ARM_CUSTMAST.FIRM_ID='01'
```

```
ARM_CUSTMAST.ZIP_CODE LIKE '92%' AND (ARM_CUSTMAST.STATE_CODE='CA' OR
ARM_CUSTMAST.STATE_CODE='OR')
```

Figure 7. Refine the query results by a) using the arrow buttons to "scroll" through the WHERE clause and b) add parentheses for grouping



Customers

Search:

Cust ID	Cust Name	Addr	Addr	City	*State	*Zip
00-0500	Ron Anderson And Company	17 Old Post Rd		Palm Springs	CA	92419
00-0300	Taylor Sporting Goods	Unit 41	1817 Augusta Cir	San Juan Capistrano	CA	92481
00-0800	Trident Industries	781 Valencia Blvd		Fullerton	CA	92516
00-1000	Mile High Bike Rentals	9833 Main St		Lake Arrowhead	CA	92547
00-0900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy		Corona Del Mar	CA	92844

Filter Wizard

Conjunction: [✓ AND] OR

Parenthesis: (Left Right)

Column:

Operator:

Value:

Save Filter

Name:

Save as filter option for all users

SQL WHERE clause

```
ARM_CUSTMAST.FIRM_ID='01'
```

```
ARM_CUSTMAST.ZIP_CODE LIKE '92%' AND (ARM_CUSTMAST.STATE_CODE='CA' OR
ARM_CUSTMAST.STATE_CODE='WA' OR ARM_CUSTMAST.STATE_CODE='OR')
```

Figure 8. With direct edit permissions, users can edit the text in the WHERE clause control

Save a Filter
Remember, once users create and test a filter to verify that it produces the desired results, they can give the filter a name and save it for future use. If security settings allow the user to create global filters, the *Save as filter for all users* checkbox is enabled, and if checked, all users will see the new filter the next time they launch the inquiry. Each saved filter appears in the listbutton at the top right of the filter form (**Figure 9**). Once saved, apply a named filter by selecting it from the listbutton, or clear filters by selecting the first (blank) row.

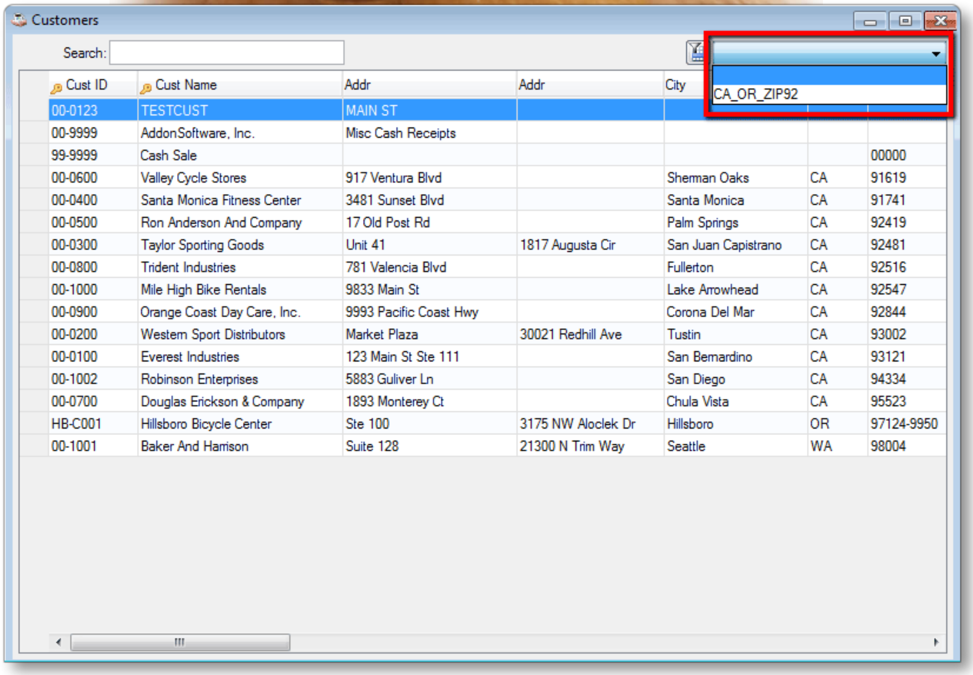
To delete saved filters, select the filter and then press [Delete] in the MDI toolbar, or press <Ctrl+D>.

New! Search all Columns
Last but not least, is a new BBj 14.0 feature, previewed in 13.10. Barista inquiry grids now offer accelerated searching with a single click. Select the *Search all columns* checkbox to look for the specified *Search* text in any column, as shown in **Figure 10**. This can be a great time saver compared to constructing a WHERE clause when looking for the same text in multiple columns. As with normal searches, the full text search honors the configuration setting for case sensitivity.

Summary

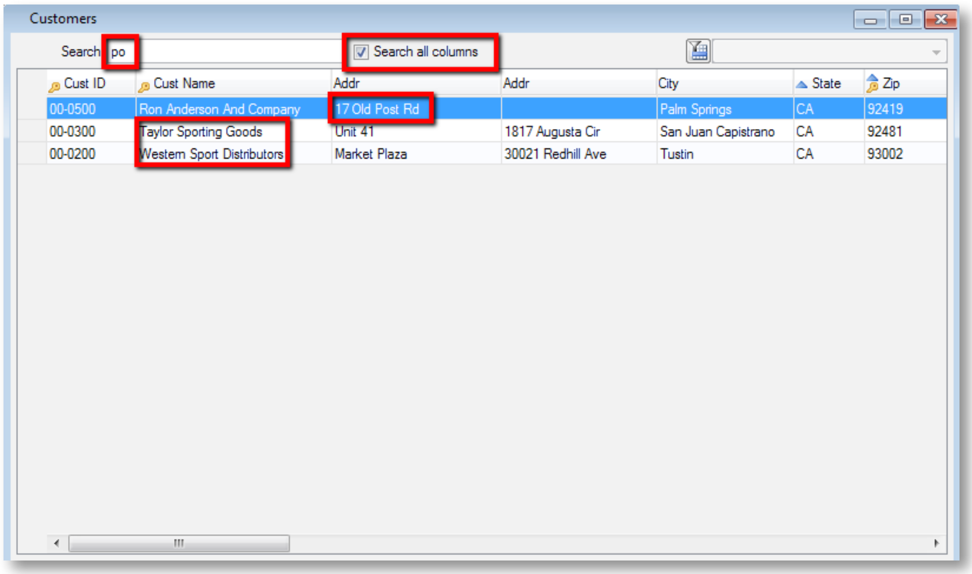
As Barista continues to evolve and gain new functionality, so do your Barista-built applications. The recent addition of advanced and flexible error handling give developers the information they need to effectively support their application. And the addition of advanced query building and editing capabilities to the often-used inquiry system means that users will be even more effective at finding the exact data they need. Finally, the text search across all columns feature will boost your users' productivity when using your Barista-designed applications. These compelling new additions add value to your new or existing application without any development effort on your part, further solidifying Barista's position as a powerful application development and runtime framework. ■

links.basis.com/**13code**



Cust ID	Cust Name	Addr	Addr	City		
00-0123	TESTCUST	MAIN ST				
00-9999	AddonSoftware, Inc.	Misc Cash Receipts				
99-9999	Cash Sale					00000
00-0600	Valley Cycle Stores	917 Ventura Blvd		Sherman Oaks	CA	91619
00-0400	Santa Monica Fitness Center	3481 Sunset Blvd		Santa Monica	CA	91741
00-0500	Ron Anderson And Company	17 Old Post Rd		Palm Springs	CA	92419
00-0300	Taylor Sporting Goods	Unit 41	1817 Augusta Cir	San Juan Capistrano	CA	92481
00-0800	Trident Industries	781 Valencia Blvd		Fullerton	CA	92516
00-1000	Mile High Bike Rentals	9833 Main St		Lake Arrowhead	CA	92547
00-0900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy		Corona Del Mar	CA	92844
00-0200	Western Sport Distributors	Market Plaza	30021 Redhill Ave	Tustin	CA	93002
00-0100	Everest Industries	123 Main St Ste 111		San Bernardino	CA	93121
00-1002	Robinson Enterprises	5883 Guliver Ln		San Diego	CA	94334
00-0700	Douglas Erickson & Company	1893 Monterey Ct		Chula Vista	CA	95523
HB-C001	Hillsboro Bicycle Center	Ste 100	3175 NW Aloclek Dr	Hillsboro	OR	97124-9950
00-1001	Baker And Harrison	Suite 128	21300 N Trim Way	Seattle	WA	98004

Figure 9. Example of a saved filter



Cust ID	Cust Name	Addr	Addr	City	State	Zip
00-0500	Ron Anderson And Company	17 Old Post Rd		Palm Springs	CA	92419
00-0300	Taylor Sporting Goods	Unit 41	1817 Augusta Cir	San Juan Capistrano	CA	92481
00-0200	Western Sport Distributors	Market Plaza	30021 Redhill Ave	Tustin	CA	93002

Figure 10. Case-insensitive search for "po" using new *Search all columns* option

For more information, refer to *Barista Error Handling* at links.basis.com/errorhandling

The Toolset of the Future

In our ongoing effort to make your BBj project development life easier and more productive, BASIS announces the new BASIS Development Tools (BDT), which now integrates with one of the most popular and powerful IDEs available, Eclipse. This article uncovers the strong evidence on which BASIS chose to move to Eclipse and provides some simple getting-started tutorials.

Goodbye NetBeans IDE (RIP)

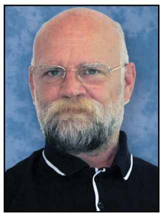
Before explaining why BASIS moved to Eclipse and taking a glimpse at the BDT, let's have a moment of silence for our BASIS NetBeans IDE.

Why are we saying goodbye to the old BASIS IDE? Here are just a few reasons:

- Newer versions of NetBeans were no longer compatible with our original architecture for BASIS plug-ins.
- Oracle/Sun refactored the NetBeans and Java infrastructure that would require a rewrite of all existing BASIS plug-ins.
- BASIS has extensive experience with Eclipse, the IDE of choice for our BASIS Java developers.

Why Eclipse?

- **It has strong roots.** In November 1998, IBM Software Group began creating a development tools platform that eventually became known as Eclipse. Industry leaders Borland, IBM, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft, and Webgain formed the initial eclipse.org Board of Stewards in November 2001, which eventually became the Eclipse Foundation.
- **It is widely used.** Eclipse has become the standard for Java-based IDEs, product development tools, modeling tools, GUI builder tools, database tools. Take a look at this 'brief' list of available Eclipse-based software at links.basis.com/esw. Eclipse is everywhere - *ubiquitous*. Eclipse released Juno on June 27, 2012 and was downloaded 1,200,000 times during its first 20 days. Of an estimated 10.3 million Java developers, 68% use Eclipse as their primary development environment.
- **It is consistent across all platforms.** Eclipse uses JFace and the Standard Widget Toolkit (JFace/SWT), an alternative to Oracle's AWT/Swing. JFace and SWT can be used independently of the rest of the Eclipse Platform, rivaling native applications written in AWT/Swing. It gives applications a polished and consistent look and feel across all platforms.
- **It is a platform.** In Eclipse's own words published on their site wiki.eclipse.org/Platform, *"The Platform defines the set of frameworks and common services that collectively make up infrastructure required to support the use of Eclipse as a component model, as a Rich Client Platform (RCP) and as a comprehensive tool integration platform. These services and frameworks include a standard workbench user interface model and portable native widget toolkit, a project model for managing resources, automatic resource delta management for incremental compilers and builders, language-independent debug infrastructure, and infrastructure for distributed multi-user versioned resource management."*
- **It is free and easy.** The platform is easily extended simply by downloading plug-ins, most of which are open source and free.
- **It is current.** The Eclipse Marketplace offers more than 6 million new and updated solutions installed directly from Eclipse. Just about every programming language there is has an Eclipse IDE available here. If you have a software problem looking for a solution, here's the place to look.
- **It is robust.** It is the Eclipse Workbench; the term Workbench refers to the entire desktop development environment, containing toolbars, editors, menus, views, etc. IBM Developerworks provides an excellent series of articles titled Mastering Eclipse v3.4. The content is dated, but the terms and names of the parts of the workbench are relevant today.
- **It is prolific.** The Eclipse Marketplace currently lists 85 different Team Development tools.
- **It is compatible.**
  - Integrates with SVN, CVS, Git, Mercurial, and Perforce for source code management.
  - Integrates with Bugzilla for bug tracking.
  - Integrates with Mylyn for task and application lifecycle management.
- **It is multi-purpose.** Eclipse can manage your databases, your team repositories, your web servers, your tasks lists, your report generators…there is a never-ending list of ways it facilitates project development of all kinds.



By Kevin Hagel
Software Developer

First Glimpse at the BDT

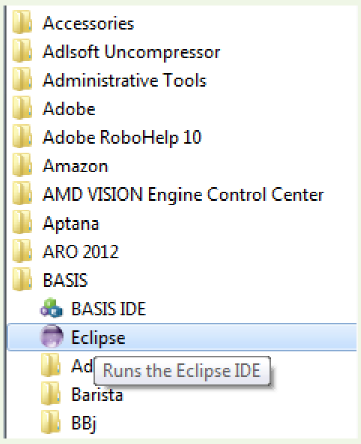
Before taking the first glimpse at BDT, let's install and run Eclipse. Even if you already have Eclipse installed, follow the **Installing the BDT** tutorial to ensure you have the correct version of Java and Eclipse before you install your BDT plug-in.

Installing the BDT

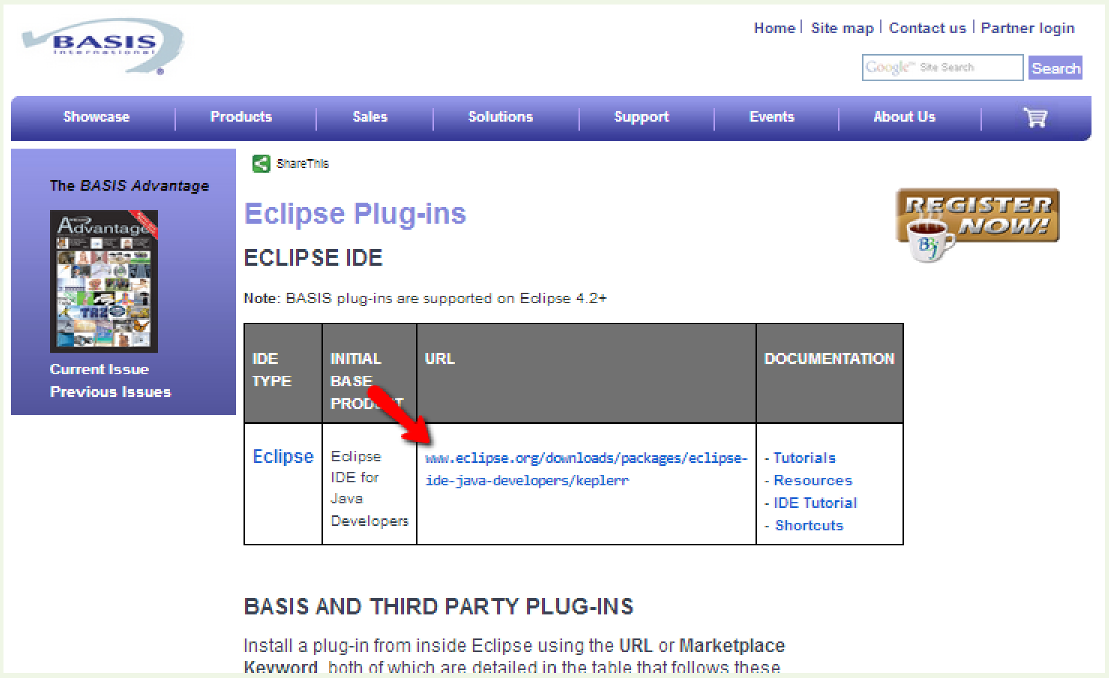
- Install Eclipse
- Determine Your Workspace
- Launch Eclipse
- Install the BDT Plug-in

Install Eclipse

1. Locate the Eclipse shortcut installed with your BBj 13.x installation, shown in the BASIS menu below.



2. Click the Eclipse icon, which opens your browser to the BASIS web page
3. Select the URL link noted below.



Eclipse Plug-ins

ECLIPSE IDE

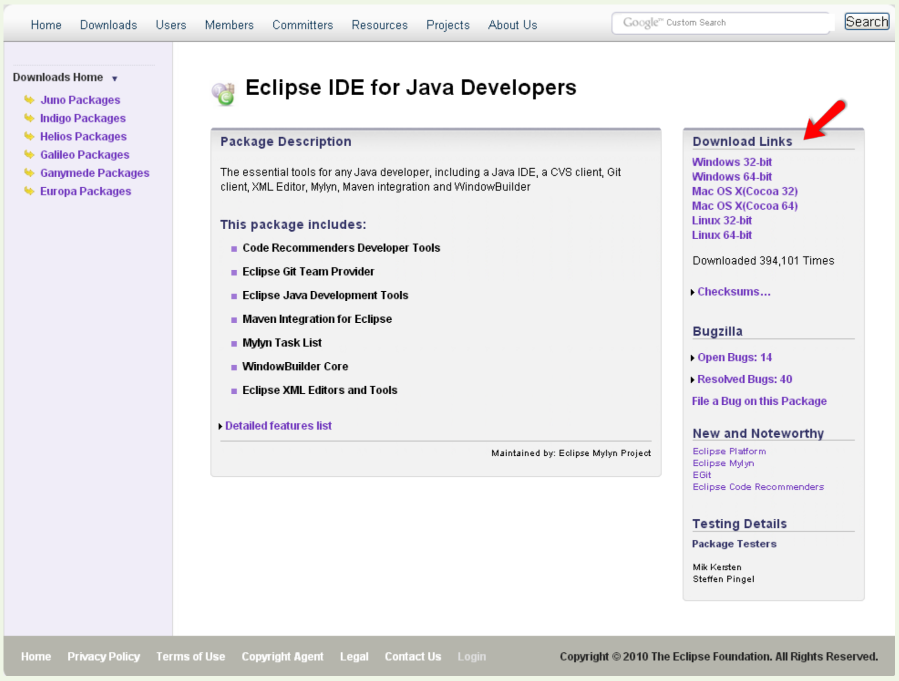
Note: BASIS plug-ins are supported on Eclipse 4.2+

IDE TYPE	INITIAL BASE PRODUCT	URL	DOCUMENTATION
Eclipse	Eclipse IDE for Java Developers	www.eclipse.org/downloads/packages/eclipse-ide-java-developers/keplerr	- Tutorials - Resources - IDE Tutorial - Shortcuts

BASIS AND THIRD PARTY PLUG-INS

Install a plug-in from inside Eclipse using the **URL** or **Marketplace Keyword**, both of which are detailed in the table that follows these

4. On the Eclipse IDE for Java Developers page, locate your operating system in the "Download Links" section to the right. As of this writing. the current Eclipse version is "Kepler," but BDT will run on Eclipse Juno or any version 4.x or higher.



Home Downloads Users Members Committers Resources Projects About Us — Google Custom Search — Search

Downloads Home
- Juno Packages
- Indigo Packages
- Helios Packages
- Galileo Packages
- Ganymede Packages
- Europa Packages

Eclipse IDE for Java Developers

Package Description

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration and WindowBuilder

This package includes:
- Code Recommenders Developer Tools
- Eclipse Git Team Provider
- Eclipse Java Development Tools
- Maven Integration for Eclipse
- Mylyn Task List
- WindowBuilder Core
- Eclipse XML Editors and Tools

Detailed features list

Maintained by: Eclipse Mylyn Project

Download Links
- Windows 32-bit
- Windows 64-bit
- Mac OS X(Cocoa 32)
- Mac OS X(Cocoa 64)
- Linux 32-bit
- Linux 64-bit

Downloaded 394,101 Times

Checksums...

Bugzilla
- Open Bugs: 14
- Resolved Bugs: 40
- File a Bug on this Package

New and Noteworthy
- Eclipse Platform
- Eclipse Mylyn
- EGit
- Eclipse Code Recommenders

Testing Details

Package Testers

Mik Kersten
Steffen Pingel

Home Privacy Policy Terms of Use Copyright Agent Legal Contact Us Login — Copyright © 2010 The Eclipse Foundation. All Rights Reserved.

NOTE: On Windows systems, we strongly advise defining an installation location with a name as short as possible and without spaces like **`C:\eclipse`** rather than **`C:\Program Files\eclipse`**. Windows can run into difficulties with long directory paths and filenames since some of the plug-ins Eclipse uses have very long file names. LINUX and Mac system do not encounter this problem.

Determine Your Workspace

Back to Installing BDT

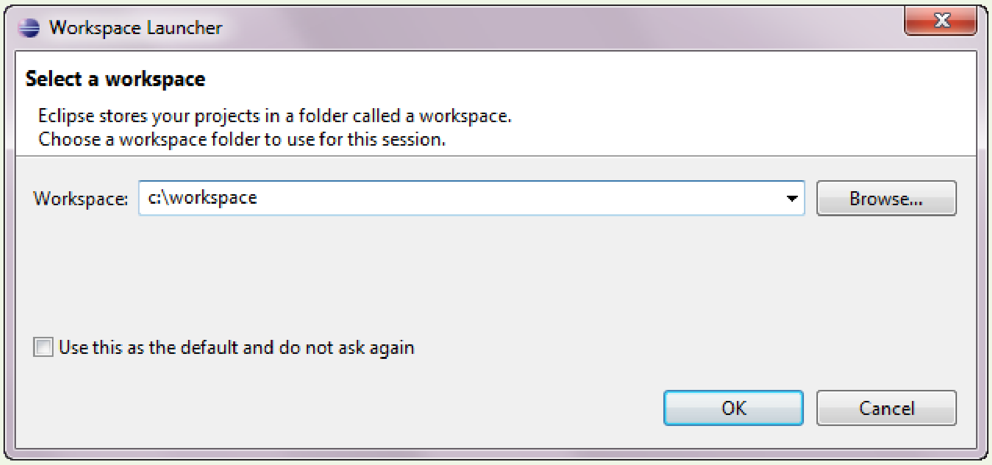
A workspace is a logical set of projects. You can have as many workspaces as you like, each with their own location and their own set of preferences and customizations.

By default project directories will be created inside the workspace directory. It is possible to create a project in an external location but still be referenced in the workspace.

Launch Eclipse

Back to Installing BDT

1. Now that you've installed Eclipse, run the executable.
2. The Workspace Launcher now appears, asking for the location of your workspace. In this case, we'll use **`C:\workspace`** shown below.



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: c:\workspace — Browse...

☐ Use this as the default and do not ask again

OK — Cancel

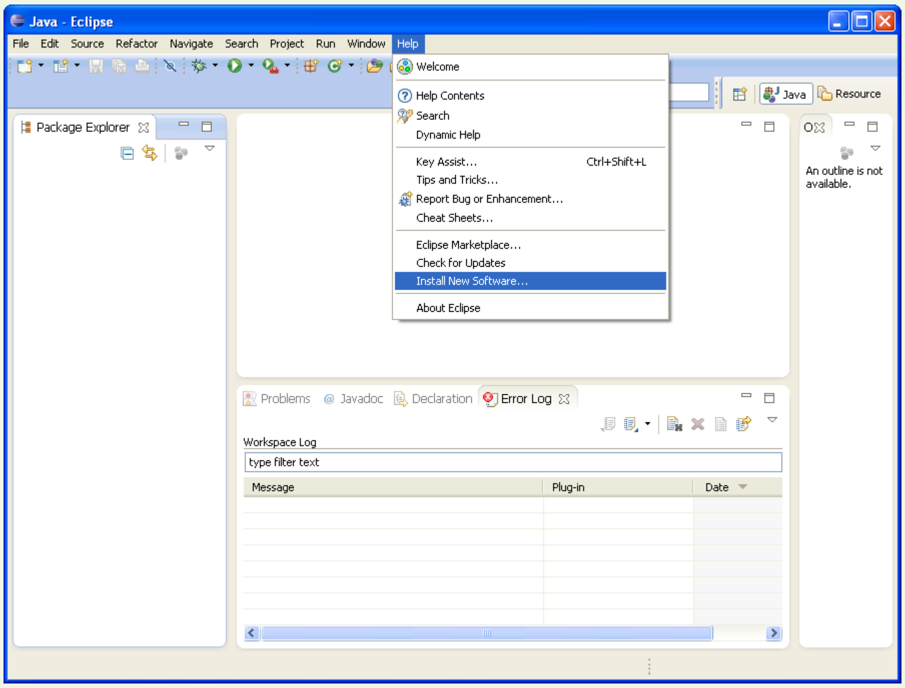
3. Optional - select the checkbox option "Use this as the default and do not ask again," if you like, and click [OK].

Install the BDT Plug-in

Back to Installing BDT

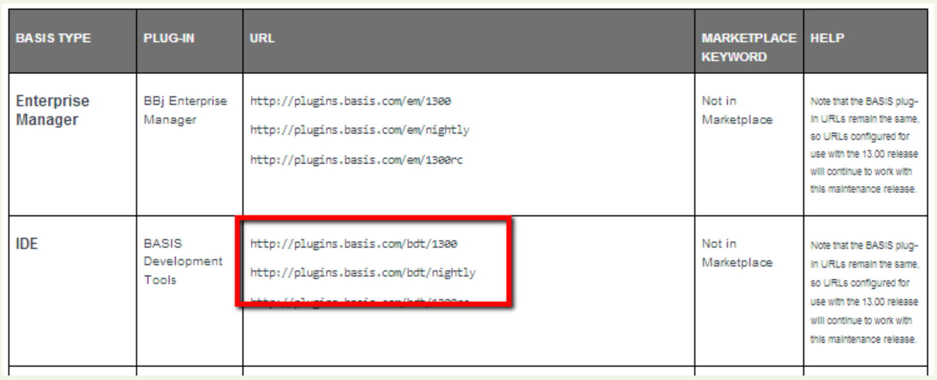
Since BDT is an Eclipse plug-in, we will install it following the same process as any other plug-in – through an "update" site located under the Help menu.

1. Select the Help > Install New Software.



You should still have the BASIS Eclipse Plug-in page open in your browser from Install Eclipse step #3. If not, locate the shortcut in your Start menu as instruction in Install Eclipse or go directly to links.basis.com/eclipse.

2. Scroll down until you locate the two BDT URLs:

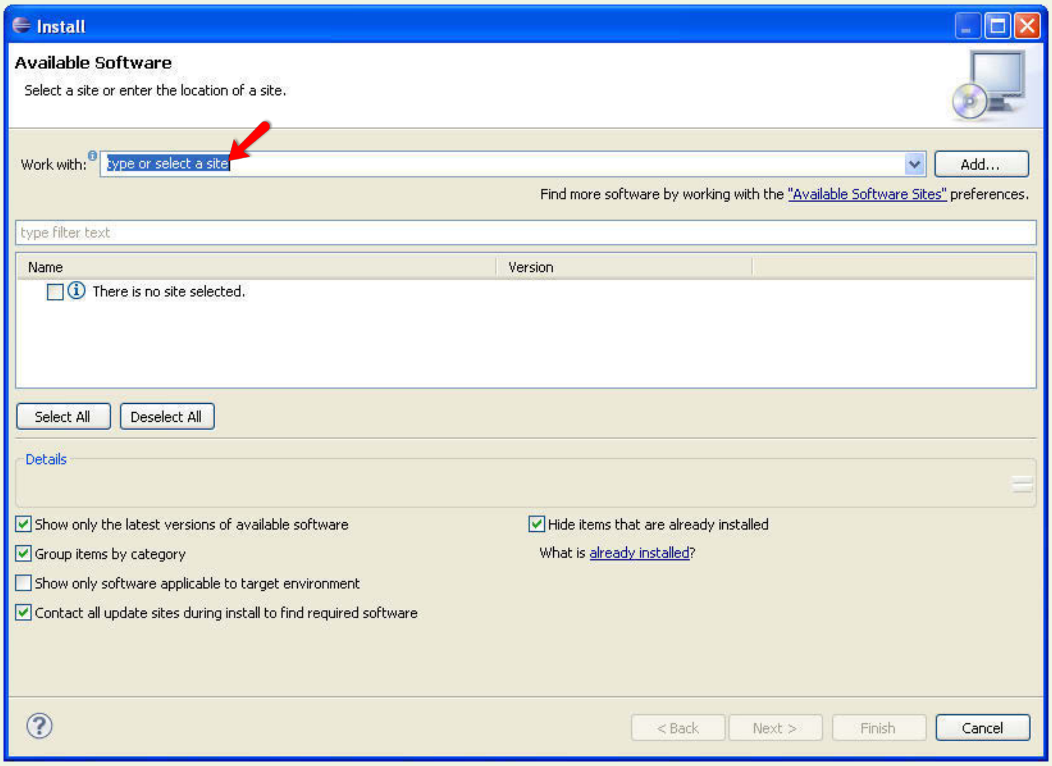


BASIS TYPE	PLUG-IN	URL	MARKETPLACE KEYWORD	HELP
Enterprise Manager	BBj Enterprise Manager	http://plugins.basis.com/em/1300 http://plugins.basis.com/em/nightly http://plugins.basis.com/em/1300rc	Not in Marketplace	Note that the BASIS plug-in URLs remain the same, so URLs configured for use with the 13.00 release will continue to work with this maintenance release.
IDE	BASIS Development Tools	http://plugins.basis.com/bdt/1300 http://plugins.basis.com/bdt/nightly http://plugins.basis.com/bdt/1300rc	Not in Marketplace	Note that the BASIS plug-in URLs remain the same, so URLs configured for use with the 13.00 release will continue to work with this maintenance release.

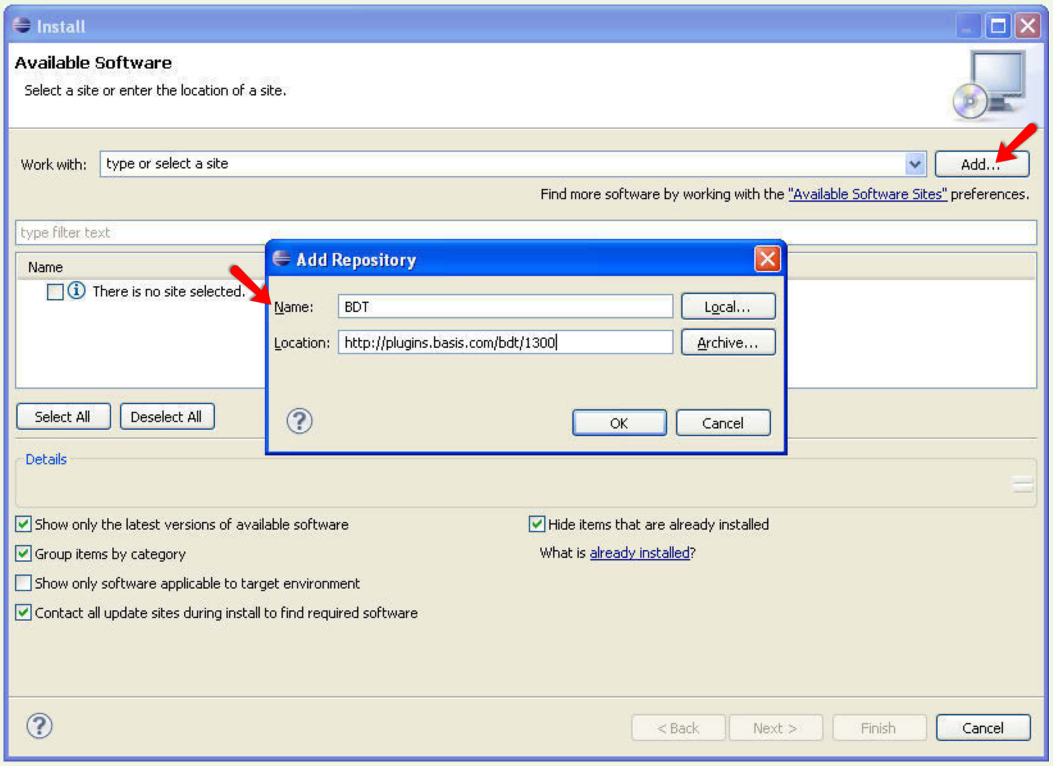
The first URL provides the latest stable release; the second (for the more adventurous), gives you the latest bleeding edge release.

3. Enter the following in the dialog window below where it says “Type or select a site”:

```
BDT - http://plugins.basis.com/bdt/1300
```


Install
Available Software
Select a site or enter the location of a site.
Work with: type or select a site
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
There is no site selected.
Select All
Deselect All
Details
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Contact all update sites during install to find required software
Hide items that are already installed
What is already installed?
< Back
Next >
Finish
Cancel

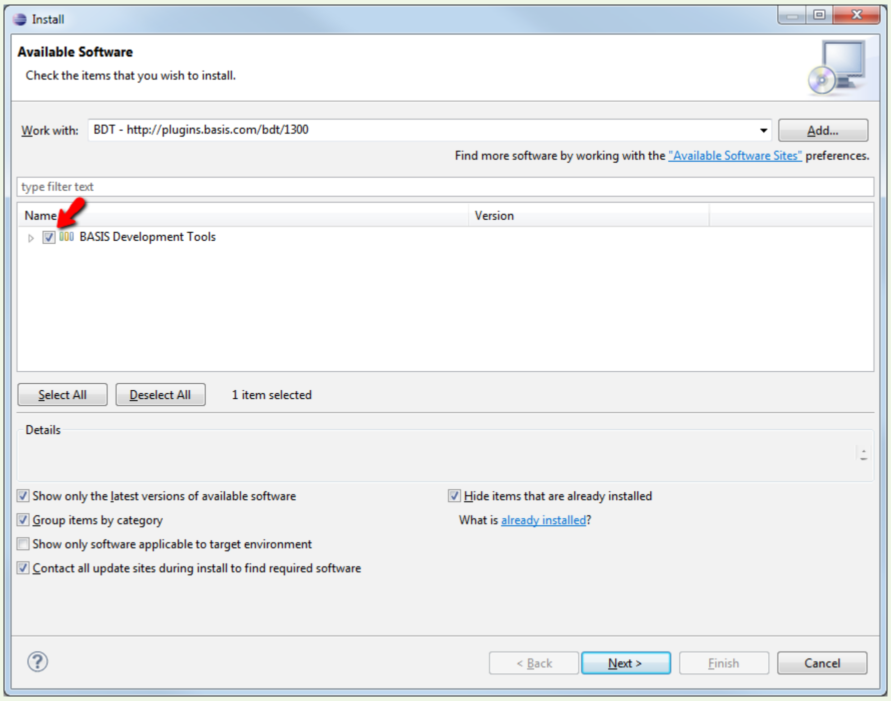
or click [Add…] and enter the same information in this way:



Name: BDT
Location: http://plugins.basis.com/bdt/1300

4. Click [OK]

5. In the dialog box shown below, mark the checkbox next to BASIS Development Tools:



Install

Available Software

Check the items that you wish to install.

Work with: BDT - http://plugins.basis.com/bdt/1300

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ BASIS Development Tools	

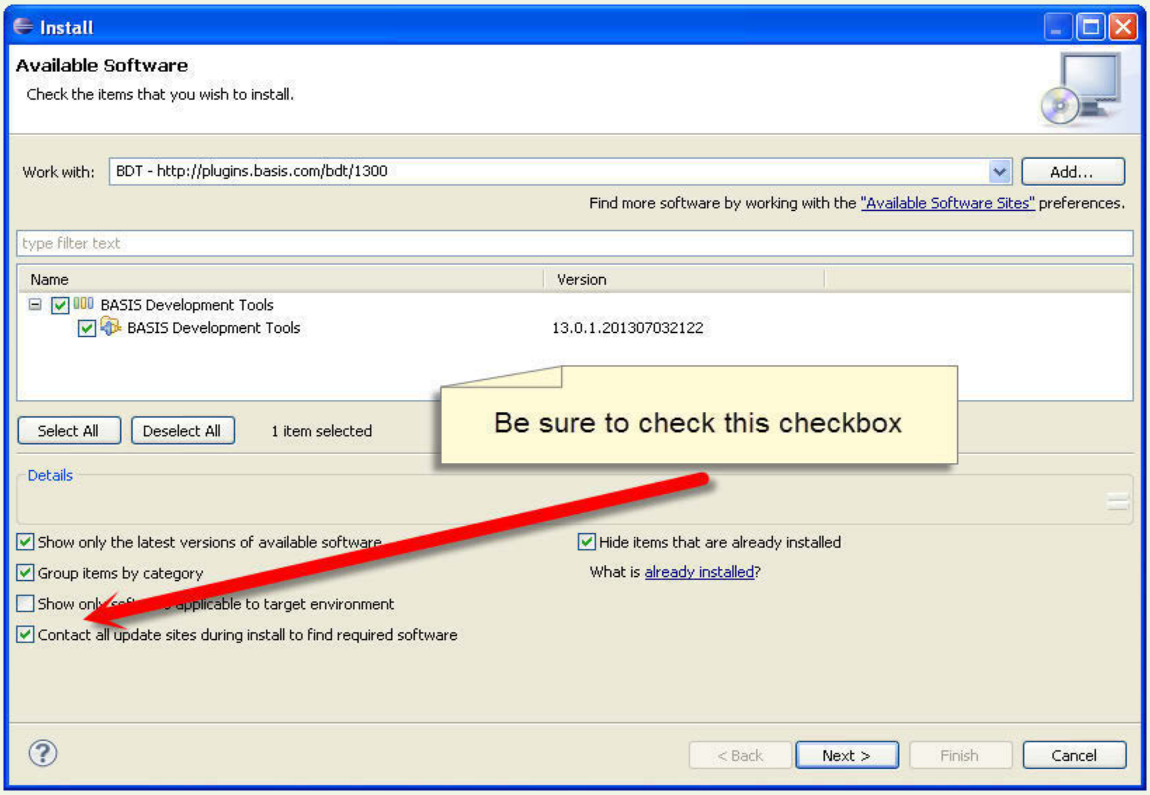
1 item selected

Details

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

6. Mark the checkbox "Contact all update sites during install to find required software."

NOTE: BDT has certain resource requirements that do not come preinstalled with Eclipse, yet are obtainable through Eclipse. If you do not select this checkbox, BDT will not install correctly. In fact, it is advisable to always mark this checkbox for every plug-in you install.



Install

Available Software

Check the items that you wish to install.

Work with: BDT - http://plugins.basis.com/bdt/1300

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ BASIS Development Tools	
☑ BASIS Development Tools	13.0.1.201307032122

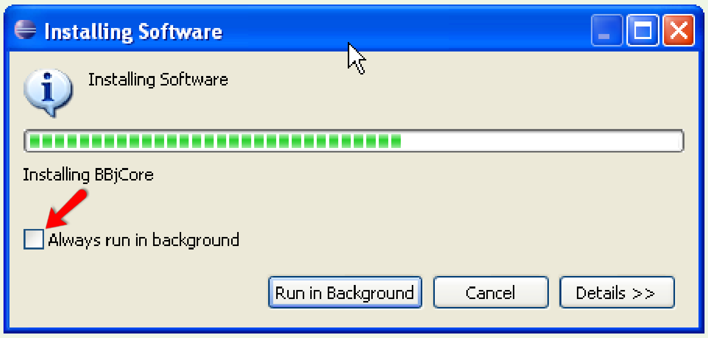
1 item selected

Details

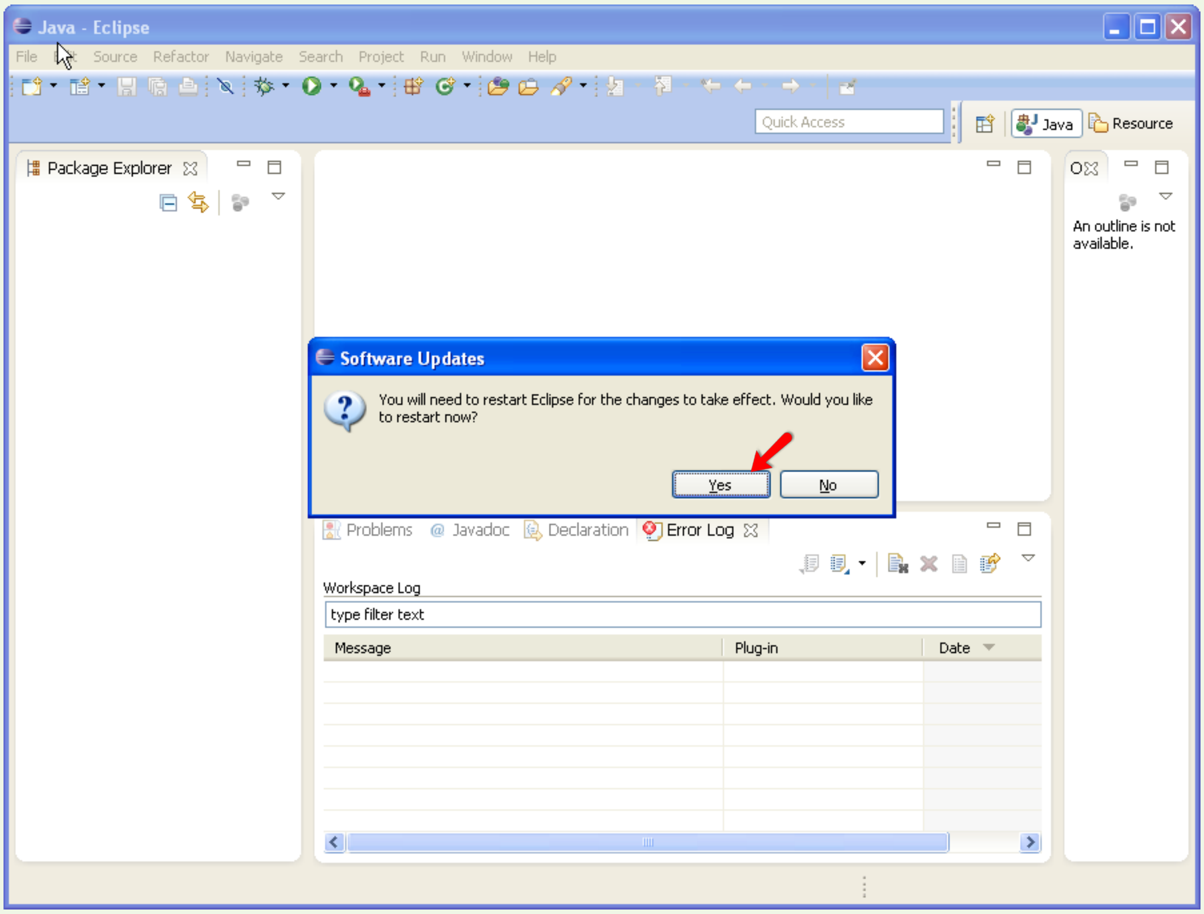
☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

7. Click [Next >] twice.
8. Select "I accept the terms of the license agreement" and press [Finish].

9. Optional - mark “Always run in background,” if desired, in the progress window.



10. Click [Yes…] once installation is complete to restart Eclipse for the changes to take effect.



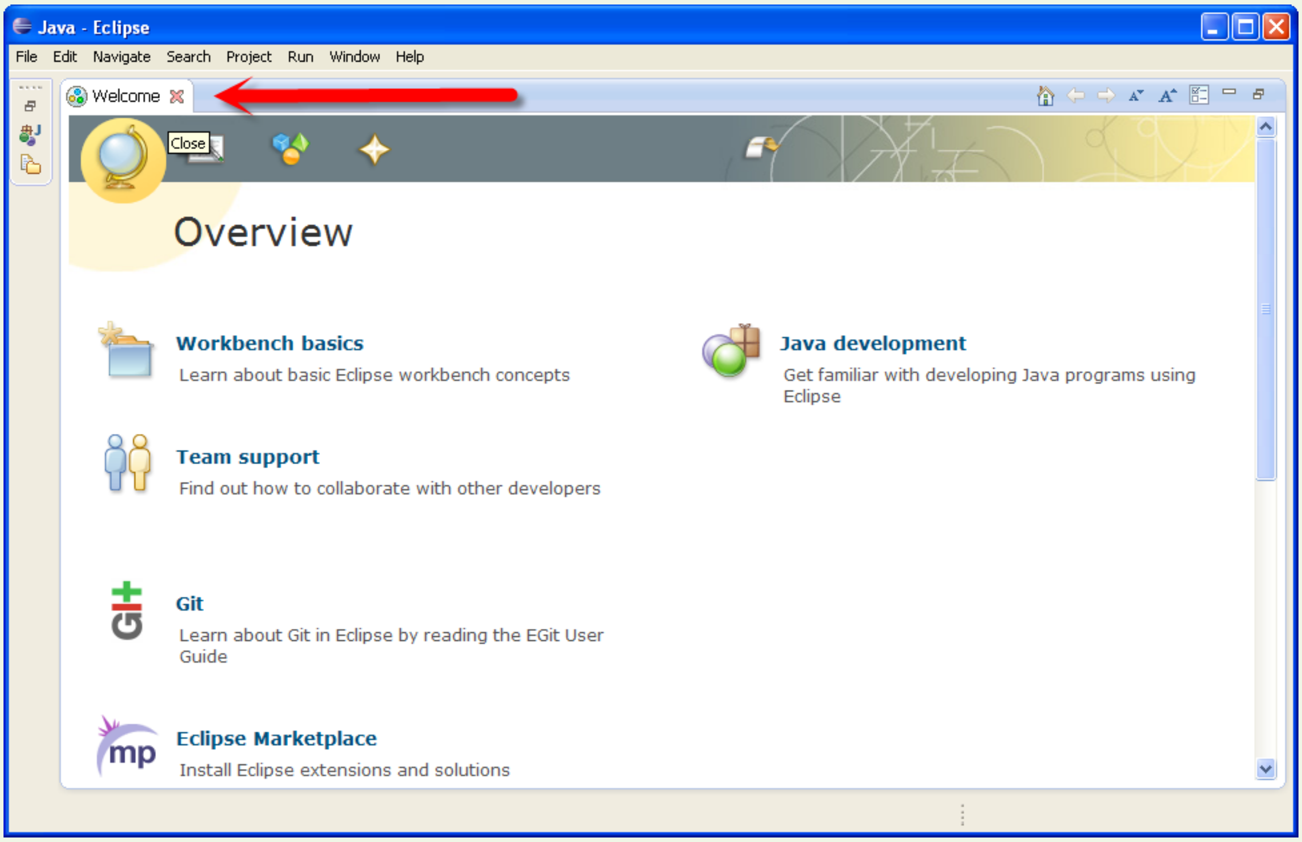
You will need to restart Eclipse for the changes to take effect. Would you like to restart now?

Eclipse will restart and once again, ask you to set your workspace.

NOTE: Eclipse will not ask for your workspace if you marked “Use this as default and don’t ask again...” in Launch Eclipse step #3.

11. Select your workspace and press [OK].

12. Optional - when Eclipse restarts, it will open with the Welcome tab that you can close by clicking the "x" shown below.


Java - Eclipse
File Edit Navigate Search Project Run Window Help
Welcome
Close
Overview
Workbench basics
Learn about basic Eclipse workbench concepts
Java development
Get familiar with developing Java programs using Eclipse
Team support
Find out how to collaborate with other developers
Git
Learn about Git in Eclipse by reading the EGit User Guide
Eclipse Marketplace
Install Eclipse extensions and solutions

Congratulations, you have installed BASIS Development Tools. The next step isn't very obvious. You will need to switch to the BASIS "perspective." Follow along in the **Guide to Perspectives**.

Guide to Perspectives

- Perspectives
- Views
- Editors
- The BASIS Perspective

Perspectives

An Eclipse "perspective" is a visual container for a set of views, menus, editors, and toolbars, available only after installing the plug-in that provides the perspective. A perspective defines the initial set and layout of views in the workbench window. Within that window, each perspective shares the same set of editors, and each perspective provides a set of functionality aimed at accomplishing a specific type of task or works with specific types of resources. Perspectives control what appears in certain menus and toolbars, defining visible action sets that you can change to customize a perspective. You can save a perspective that you built in this manner so that you can open it again later.

Some common perspectives you may use are BASIS, BBj Enterprise Manager, Debug, SVN Repository, Java, Report Design, Resource…many more. Typically a plug-in provides its own perspective.

You can customize a perspective by selecting **`Window > Customize Perspective`** and then choosing which toolbars appear, what appears in them, the contents of the views, etc. Then just save this customized perspective with your own chosen name.

Views

Views support editors and provide alternative presentations as well as ways to navigate the information in your workbench. For example, the BASIS Navigator and Project Explorer and other navigation views display projects and other resources that you are working with.

Views also have their own menus. Some views also have their own toolbars. The actions represented by buttons on view toolbars only affect the items within that view. A view might appear by itself, or stacked with other views in a tabbed notebook.

To change the layout of a perspective, open and close the views and by docking them in different positions in the workbench window. In addition, you can detach views from the workbench outside the bounds of the IDE to take advantage of multiple monitors.

Editors

Back to Guide to Perspectives

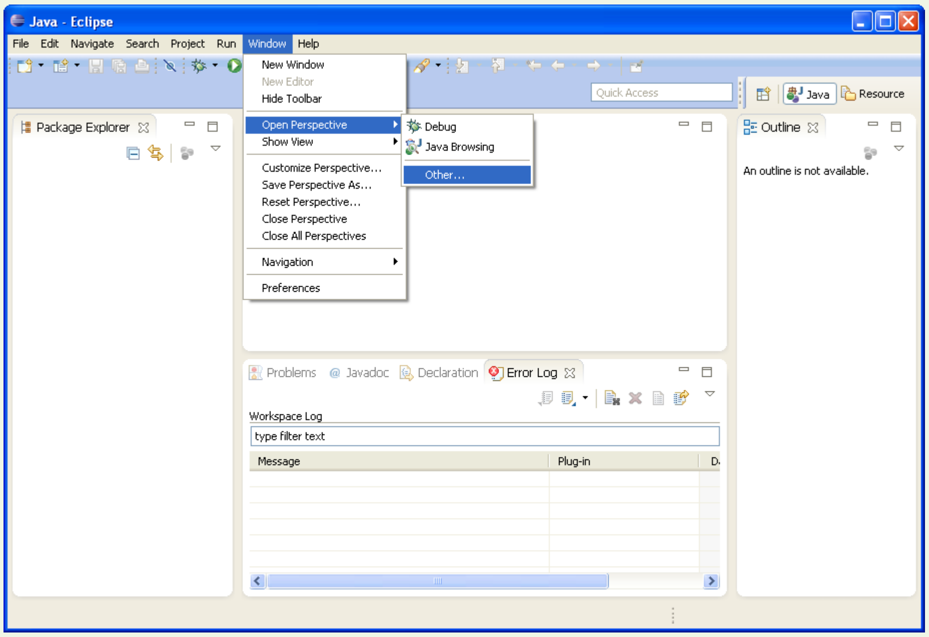
Most perspectives in the workbench are comprised of an editor area and one or more views. Any number of editors can be open at once, but only one can be active at a time. The main menu bar and toolbar for the Workbench window contain operations that are applicable to the active editor. Tabs in the editor area indicate the names of resources that are currently open for editing. An asterisk (*) indicates that an editor has unsaved changes. By default, editors are stacked in the editor area, but you can choose to tile them in order to view source files simultaneously.

The BASIS Perspective

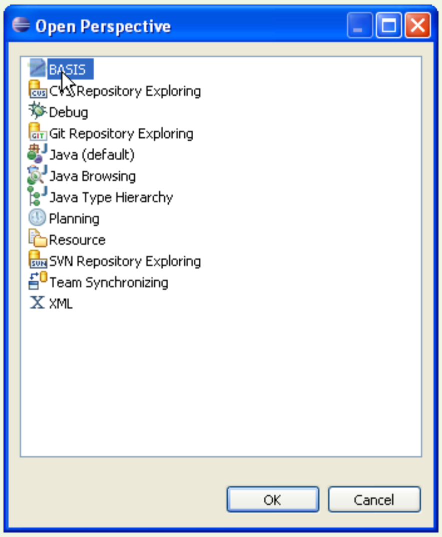
Back to Guide to Perspectives

To open the BASIS perspective,

1. Choose Window > Open Perspective > Other...



2. Double-click on BASIS or select BASIS in the Open Perspective window, and press [OK].



Now you have the BDT's BASIS Perspective open and selected. Let's go over some of the component parts.

BDT Components

- Projects, Folders, and Files
- BASIS Navigator
- Editor
- Builder
- Executor
- Debugger

Projects, Folders, and Files

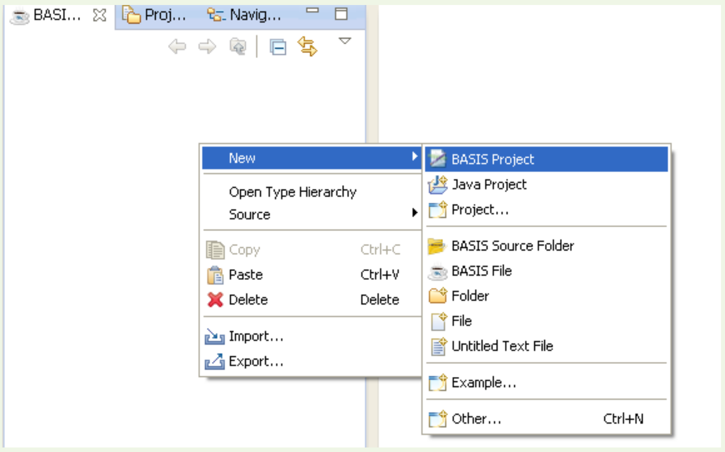
There are three different types of resources in the workbench: projects, folders, and files. Projects are the largest structural unit used by the workbench. Projects contain folders and files, and developer can open, close, or build them. Folders can contain other folders and files. The workbench and BDT itself provide a number of mechanisms for working with projects, folders, and files.

BDT projects contain source folders, output folders, source modules (files, as in “Foo.bbj”), tokenized files, and related configuration files for building a BBj program. BDT projects can also contain “foreign resources,” which typically are image files, html, Jasper configurations, etc. Source modules pass through the compiler to produce tokenized files. All foreign resources are simply copied into the output folder(s).

Let’s illustrate these concepts by creating a simple BASIS project.

Create a project

1. Select from the menu at top, File > New > BASIS Project, or right-click in the BASIS Navigator and select New > BASIS Project.



2. Create new project as follows:

a. Project name: Demo

b. Accept all the defaults for the new project configuration for:

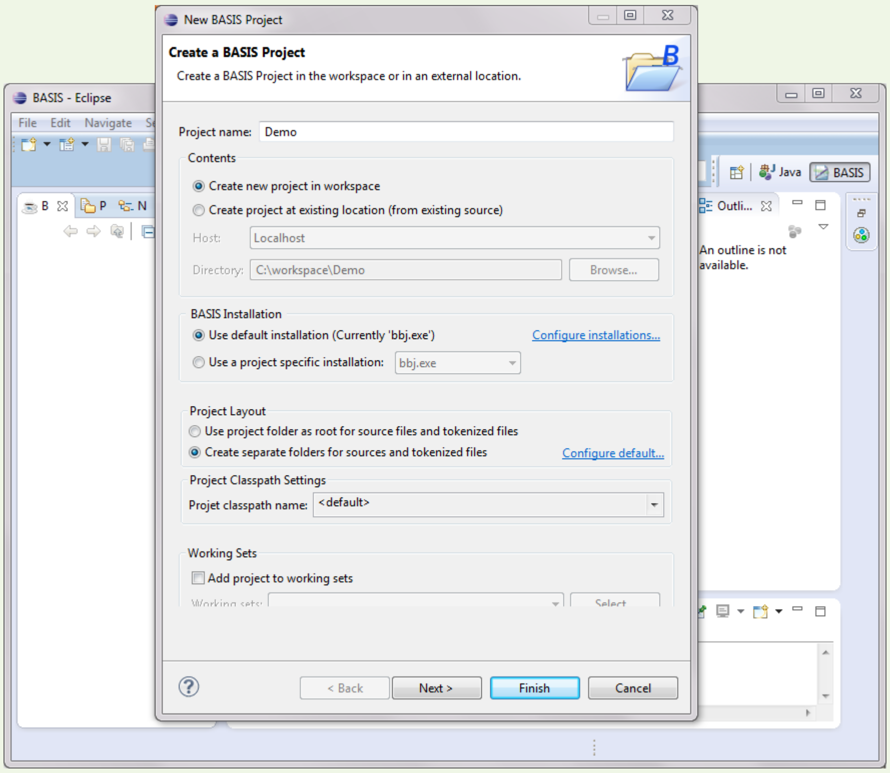
i. Contents - Create the new project in workspace. If you originally chose a workspace name `C:\workspace`, this creates a new directory `C:\workspace\Demo`.

ii. BASIS Installation - Use default installation. This sets up your project to use the default BASIS installation resources for your project.

iii. Project Layout - Create separate folders for sources and tokenized files. By default this will create two subdirectories in your project `C:\workspace\Demo\src` to contain the BBj source files and foreign resources, and `C:\workspace\Demo\bin` to contain the tokenized files and copied foreign resources

iv. Project Classpath Setting - Leave as default for this tutorial.

v. Working Sets - Leave as default for this tutorial.



An outline is not available.

New BASIS Project

Create a BASIS Project

Create a BASIS Project in the workspace or in an external location.

Project name: Demo

Contents

(●) Create new project in workspace
( ) Create project at existing location (from existing source)
Host: Localhost
Directory: C:\workspace\Demo Browse...

BASIS Installation

(●) Use default installation (Currently 'bbj.exe') Configure installations...
( ) Use a project specific installation: bbj.exe

Project Layout

( ) Use project folder as root for source files and tokenized files
(●) Create separate folders for sources and tokenized files Configure default...

Project Classpath Settings

Projet classpath name: <default>

Working Sets

☐ Add project to working sets
Working sets: Select

< Back Next > Finish Cancel

3. Click [Finish]. The New BASIS Project wizard closes and reveals your project in the BASIS Navigator.

4. In the BASIS Navigator,
- **a.** Select the Demo project to expand the project and reveal the current project contents.
- **b.** Right-click on the src folder.
- **c.** Select New > BASIS File.
- **d.** Type foo as the file name in the dialog that appears.

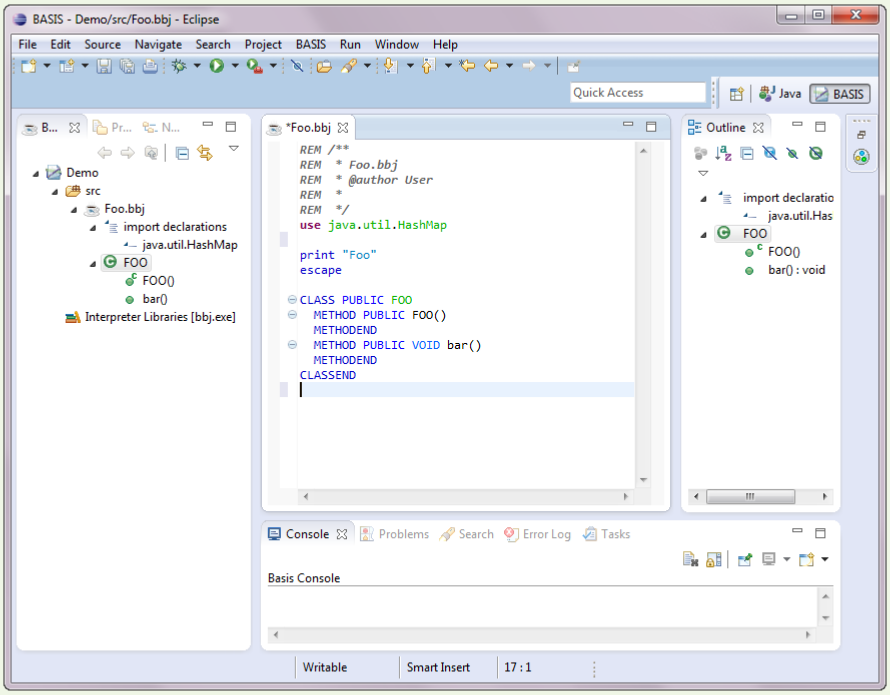
The wizard now creates Foo.bbj and opens it in the editor, using the preconfigured source file template. Next, we are going to add a simple class with some simple methods to help us see what the BASIS Navigator can do.

Add a Class

1. Enter the following source so that your file Foo.bbj contains the following ...

```
REM /**
REM * Foo.bbj
REM * @author User
REM *
REM */

use java.util.HashMap
print "Foo"
escape

CLASS PUBLIC FOO
 METHOD PUBLIC FOO()
 METHODEND
 METHOD PUBLIC VOID bar()
 METHODEND
CLASSEND
```

… and you should see the BASIS Navigator and new Demo project in the left pane and other information in the following:



```
REM /**
REM  * Foo.bbj
REM  * @author User
REM  *
REM  */
use java.util.HashMap

print "Foo"
escape

CLASS PUBLIC FOO
  METHOD PUBLIC FOO()
  METHODEND
  METHOD PUBLIC VOID bar()
  METHODEND
CLASSEND
```

Back to BDT Components

BASIS Navigator

The far left pane contains three tabs: the BASIS Navigator, the Project Explorer, and the Resource Navigator. The BASIS Navigator presents a project-level view of your project in a tree structure, representing the BDT project model. This model starts at the top with the source folder, the source module (Foo.bbj), then down to the internals of the source module – the import declarations (your use statement), class declarations, field declarations, method declarations, and so on.

To navigate to the various parts of your source in the editor, double-click on the element in the navigator. If you press the Link with Editor icon (the two arrows at the top of the navigator, then you can navigate in the editor by simply clicking on the element. This isn't such a magical feature with a tiny source file like this, but imagine a very large file of 10-20 classes, many methods per class, and 1000+ lines of code…this is where navigator really comes in handy!

You can also create new source folders here. For example, you want to keep your .html or your images or other file types in a separate directory. Simply select Demo in the Navigator, File > New > BASIS Source Folder and name it: html. Then create a second source folder and name it: images. To place image files in this new folder, you can import them or drag them into the images folder from your File Explorer (or whatever your OS calls it).

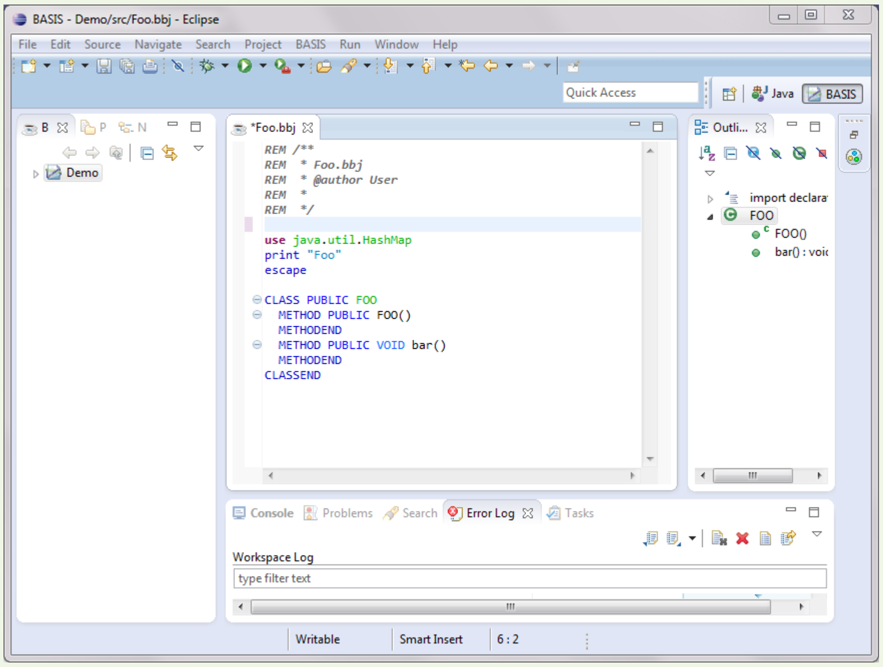
It is possible to have all your source in the project root directory, but it is considered best practice to always put your sources in folders underneath the project. Keep in mind this has consequences in how you Build, Execute, and Debug your project.

Back to BDT Components

Editor

The BASIS Editor is just one of many different editors available in the Eclipse workspace. Different editors are associated with different types of files. For example, the Java editor is associated with files matching *.java. You may have default editors for .xml files, or special-purpose xml editors if you've downloaded plug-ins for them. All BBj source files open in the BASIS Editor, by default. Simply open an editor by double-clicking on the file in the Navigator.

You can open any number of editors at the same time, but only one can be active at any one time. The main menu bar and toolbar for the workbench window contain operations that are applicable to the active editor. Tabs in the editor area indicate the names of resources that are currently open for editing. An asterisk (*) indicates that an editor has unsaved changes.



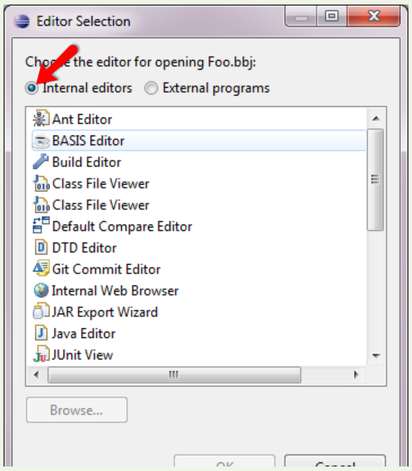
```
REM /**
REM  * Foo.bbj
REM  * @author User
REM  *
REM  */

use java.util.HashMap
print "Foo"
escape

CLASS PUBLIC FOO
  METHOD PUBLIC FOO()
  METHODEND
  METHOD PUBLIC VOID bar()
  METHODEND
CLASSEND
```

The gray border at the left margin of the editor area may contain icons that flag errors, warnings, or problems the system detects. Icons also appear if you have created bookmarks, added breakpoints for debugging, or recorded notes in the Tasks view. You can view details for any icons in the left margin of the editor by hovering the mouse over them.

Sometimes you may need to use an external editor to modify a file in the workbench. This can occur if you don't have an editor registered for a certain file type, or if you wish to use a particular external editor to modify a file in the workbench even if you do have an internal editor registered for that file type.

Use an Internal Editor

1. Select the file in the Navigator.
2. Right click on the file and choose Open With > Other…
3. Check the "Internal editors" radio button (or "External programs" to select an external editor).



Cho… the editor for opening Foo.bbj:
[✓ Internal editors] External programs

- Ant Editor
- BASIS Editor
- Build Editor
- Class File Viewer
- Class File Viewer
- Default Compare Editor
- DTD Editor
- Git Commit Editor
- Internal Web Browser
- JAR Export Wizard
- Java Editor
- JUnit View

4. Select the Internal editor of your choice.
5. Click [OK].

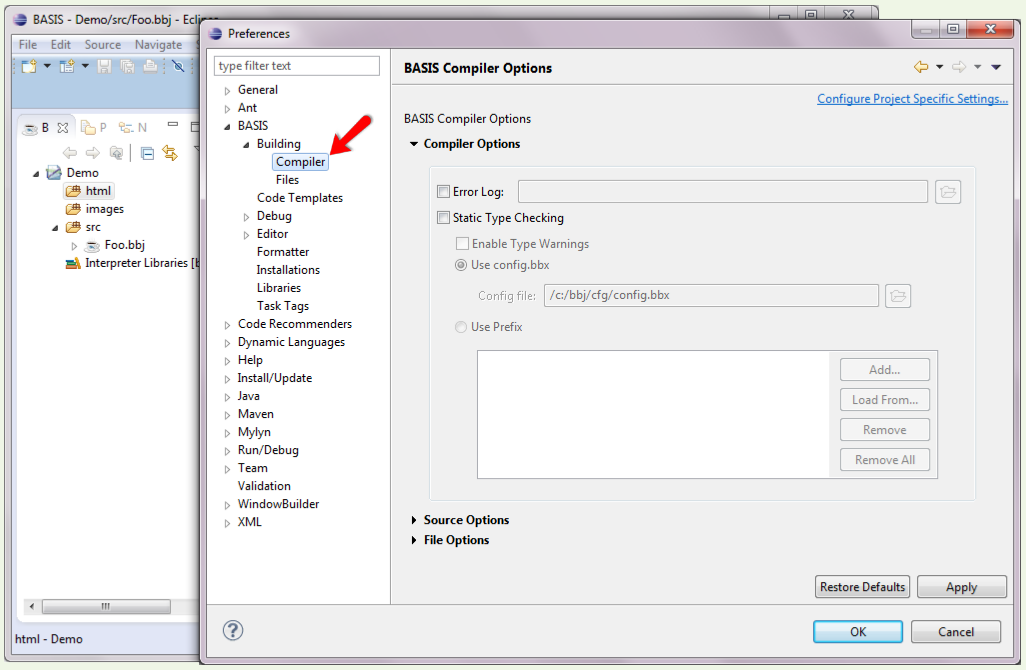
Back to BDT Components

Builder

The BDT Builder is what you use to build your BASIS projects. There are many options to support the kind of project build you'd like. Do you want to tokenize files? What options do you want to pass to the bbjcpl compiler? Where do you want your tokenized files to go? What kind of type checking do you want performed during the build?

We need to use the Eclipse Preferences dialogs to see how you can configure your Build options and Compile options for the project.

Configure Build and Compile
1. In Windows and Linux, select Window > Preferences; on the Mac, select the Eclipse > Preferences.
2. Under Preferences, find and expand the BASIS node (aka Preferences > BASIS).
3. Click on Building > Compiler.



BASIS Compiler Options

BASIS Compiler Options
Compiler Options
☐ Error Log:
☐ Static Type Checking
☐ Enable Type Warnings
◉ Use config.bbx
Config file: /c:/bbj/cfg/config.bbx
○ Use Prefix

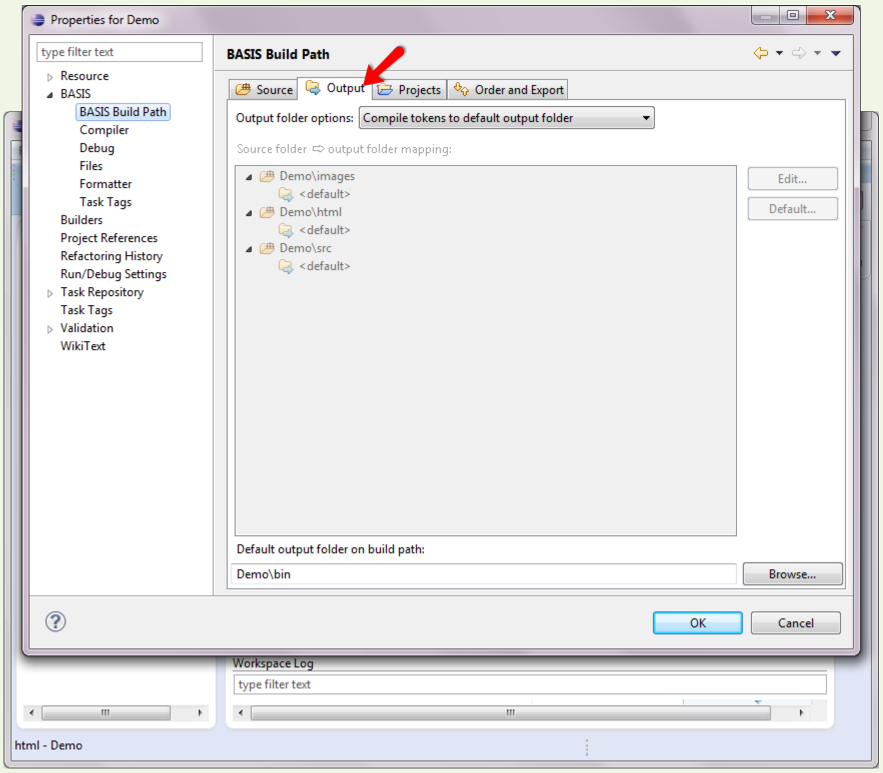
Set Compiler Options

Refer to the topic bbjcpl - BBj Compiler in the online help. In this dialog, we can set the default command line options to send to bbjcpl to build each file.

These are workspace-wide settings. If you prefer to override them and set project-specific settings for individual projects in your workspace, click the Configure Project Specific Settings... link in the top right corner of this dialog window.

There's more to it, however; you need to choose how to build the entire project.

Build the Project
1. Select Demo in the Navigator.
2. From the project menu at top, select Project > Properties > BASIS > Compiler. Here you see the same dialog as in previously appeared, giving you the option now to modify the default compiler options or set a project specific compiler options settings.
3. Select BASIS Build Path and then select the Output tab shown below.



Properties for Demo

type filter text

- Resource
- BASIS
  - BASIS Build Path
  - Compiler
  - Debug
  - Files
  - Formatter
  - Task Tags
- Builders
- Project References
- Refactoring History
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

BASIS Build Path

Source | Output | Projects | Order and Export

Output folder options: Compile tokens to default output folder

Source folder ⇨ output folder mapping:

```
Demo\images
    <default>
Demo\html
    <default>
Demo\src
    <default>
```

Edit... Default...

Default output folder on build path: Demo\bin

Browse... OK Cancel

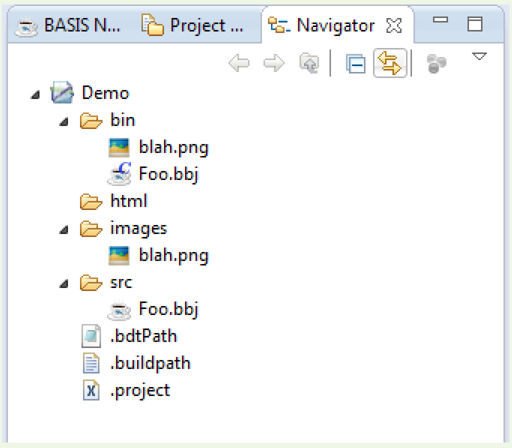
Workspace Log

html - Demo

We created our Demo project with the default project options, which include the option to Compile tokens to the default output folder. This default output folder “bin” does not appear in the BASIS Navigator, but this is where all the tokenized files are written, with respect to any subfolders they might appear in. Since we created the images and html subdirectory, you can see they are all set to also compile to default output folder. Non-source files will not be compiled of course; they will simply be copied into the bin folder.

It is possible to have multiple output folders as well, however, we will keep the defaults for this purpose.

> **Hint** - if you would like to see the ‘hidden’ directories and files, go to the navigators section at left, select the Navigator tab. See the screenshot below of the project fully expanded to show all the parts. This navigator, commonly called the Resource Navigator, is very much like a directory listing view of your project.



BASIS N... | Project ... | Navigator

```
Demo
    bin
        blah.png
        Foo.bbj
    html
    images
        blah.png
    src
        Foo.bbj
    .bdtPath
    .buildpath
    .project
```

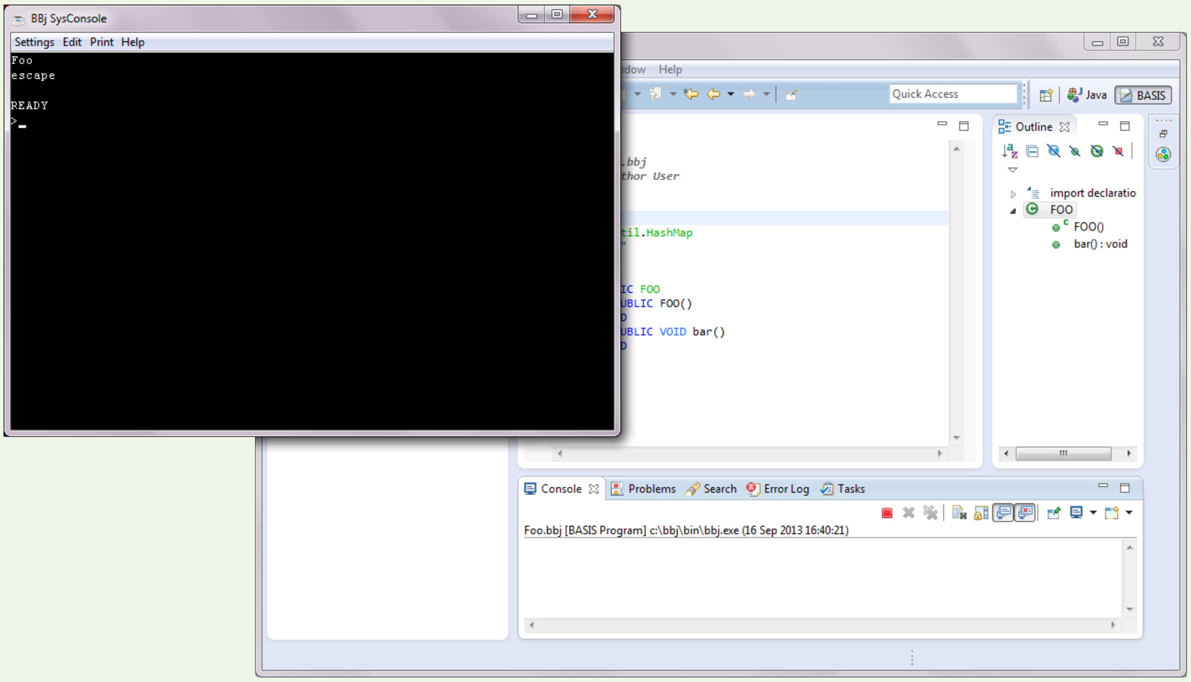
In this image, you can see the bin folder. You can see that your file Foo.bbj has been compiled into Foo.bbj - notice the little blue ‘C’ above and to the right of the file in the bin folder. This indicates a Compiled file. Notice also that our image file blah.png copied into the bin folder from its location in the images source folder.

Back to BDT Components

Executor

Execute a file in Eclipse through a Launch Configuration. To create a default launch configuration, follow the steps below.

1. Select Demo in the BASIS Navigator (not in the Resource Navigator) using the Run menu; select Run > Run As > BASIS Project to display what is shown below.



```
Foo
escape

READY
>_
```

```
Foo.bbj [BASIS Program] c:\bbj\bin\bbj.exe (16 Sep 2013 16:40:21)
```

2. Type **bye** at the Ready prompt in the BBj SysConsole to exit the program.

Of course, this was a very simple BBj program and project. Since we only have one source module in this project, BDT knows to create a Launch Configuration for that file only. If you have more than one source file, BDT will ask which file you want to run.

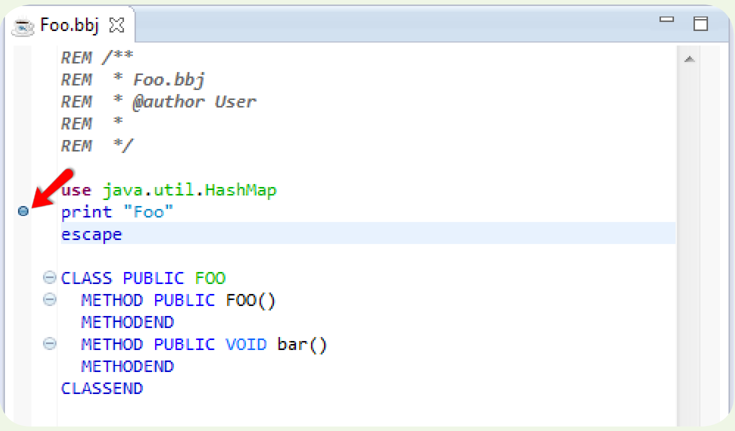
Back to BDT Components

Debugger

Debugging a file is launched the same way as running a file, through Launch Configurations. In fact, you can use the same launch configuration you created to Run > Run As > BASIS Program.

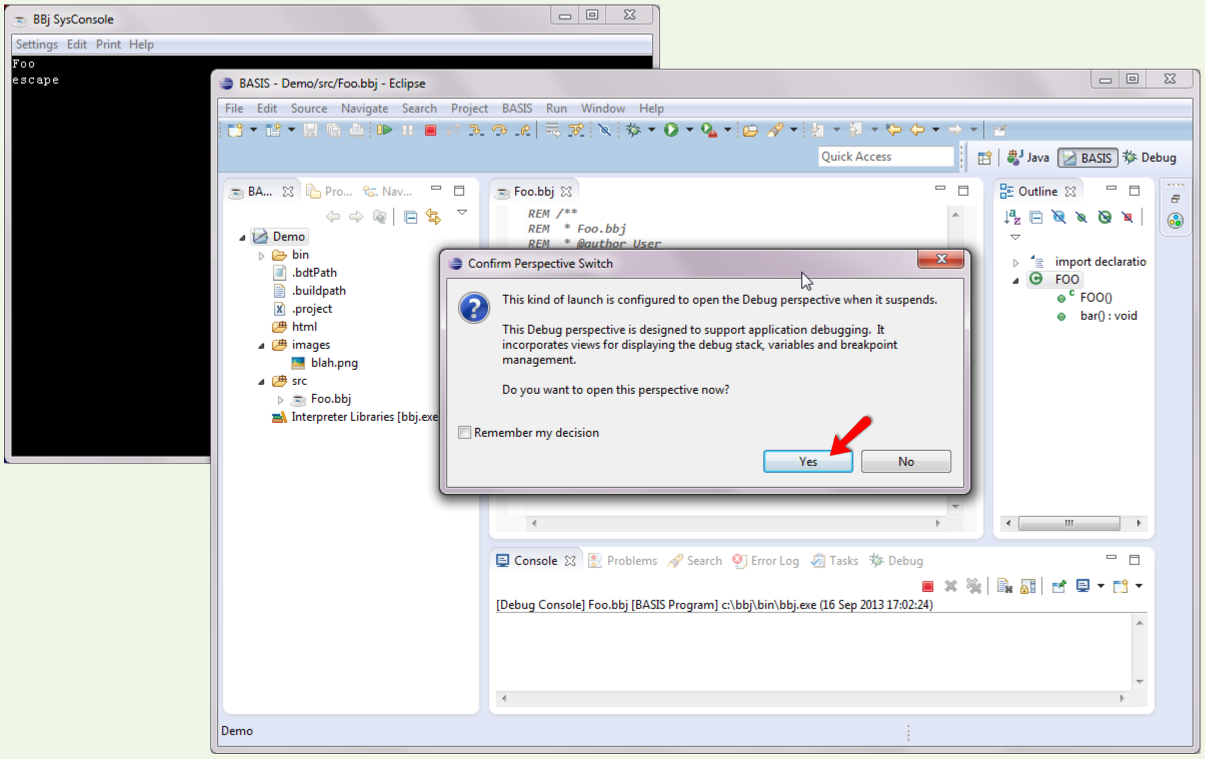
First, let's set a breakpoint on the line **`print "Foo"`** in Foo.bbj.

1. Move your mouse cursor into the gray column just to the left of that line and double-click. The small blue ball in that column indicates a breakpoint is set on that line.



```
REM /**
REM  * Foo.bbj
REM  * @author User
REM  *
REM  */

use java.util.HashMap
print "Foo"
escape

CLASS PUBLIC FOO
  METHOD PUBLIC FOO()
  METHODEND
  METHOD PUBLIC VOID bar()
  METHODEND
CLASSEND
```

2. Now select the project, Run > Debug As > BASIS Program.
3. Click [Yes] to confirm you want to switch to the Debug perspective.



```
Foo
escape
```

```
REM /**
REM  * Foo.bbj
REM  * @author User
```

Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

☐ Remember my decision

Yes No

```
[Debug Console] Foo.bbj [BASIS Program] c:\bbj\bin\bbj.exe (16 Sep 2013 17:02:24)
```

You will then switch to the Debug perspective and see the line highlighted where the breakpoint hit.

4. Press [F8] to continue running.

Summary

The BASIS Development Tools, tightly integrated as a plug-in to the ever-popular Eclipse IDE, is leap years ahead of its NetBeans-based predecessor. Regardless of which platform you develop on, Eclipse looks good and offers a host of existing plug-ins and configurability options to optimize and streamline your development cycle. If you have not tried it yet, now is the time. Find out how truly powerful and integrated this development environment really is! 🟩

For additional assistance, refer to these resources
- Online Juno Help System
- Online Kepler Help System
- Eclipse IDE Tutorial
- Vogella Eclipse Tutorials
- IBM developerWorks
- Eclipse Tutorial on sourceforge
- Eclipse Resources

A Unique Business Opportunity

While one source dates the familiar proverb *"Opportunity knocks but once"* back to 8 A.D., the saying still holds true today. Ignoring opportunities can result in great remorse. One very important opportunity is knocking loudly on your door and likely, one of the following scenarios apply to you.

Which Scenario Best Describes You?

As a company owner, business development manager, or team leader...

- I have my own custom ERP solution and it's time to modernize it.

  Why not use out-of-the-box ERP system modules?

  Does adding third party utilities drastically reduce your actual margin? Get a bigger slice of a bigger margin pie by eliminating them.

- My sales volumes determine my margins with my OEM supplier but I deserve more.
- I am successful with my current ERP product set but waste valuable time with the following:
  - Updating non-differentiating modules of my ERP solution, like General Ledger
  - Redoing customer modifications with every new release of the product
  - Supporting multiple versions of my solution because my customer's modifications have left them "stuck-in-time"
- My current product line is tied to one particular operating system and/or can't be deployed in the cloud
- My current product line does not scale well.

  Technically, it can't accommodate large user counts.

  It is too costly for businesses with small user counts.

- I have a ton of good ideas for the product but lack the resources to implement them.

As an individual...

- I want to make more money using my IT know-how.
- I work hard and know I can run a successful business if I just had the opportunity but I can't afford the start-up costs.

If one of these scenarios described you, you're the missing piece!

Read on...

By Paul D. Yeomans
Vertical Market Account Manager

The Basics of the Solution

AddonSoftware® by Barista® is a proven and trusted solution with many years in the market, refined and enhanced through its use. This solution scales extremely well from a small handful of users up to hundreds of users. It is priced competitively, platform independent, and provides you with deployment choices including on-premise or hosted cloud. It is flexible. Since your deployment choice uses the same code, you can change deployments later. Move to the cloud and back as your requirements change.

It is simple to sell. AddonSoftware is ready to deploy and includes the source code along with the RAD tool, Barista Application Framework. You just have to determine the number of concurrent ERP seats required. Keeping it easy, we distribute AddonSoftware in three bundles:

- **Accounting** includes Accounts Payable, Accounts Receivable, and General Ledger
- **Distribution** includes the Accounting bundle + Inventory Control, Sales Order Processing, Purchase Order Processing, and Sales Analysis
- **Manufacturing** includes both the Accounting and Distribution bundles + Bill of Materials and Shop Floor Control

More Reasons to Consider AddonSoftware

We own, develop, and support the full technology stack, from the language and development tools to the ERP solution. Are you paying for installers, licensing, report writers, and email/fax functionality? You no longer need those extra expenses. By building in most of those utilities you would otherwise purchase from a third party, we save you and your customer money that improves your margins.

It is Easy to Get Started

With no upfront financial investment for the entry partner level, cost barriers common in the industry are non-existent. Just bring your vertical knowledge and development skills along with your business drive. We supply you with the tools, training, and a support framework to lay a foundation for your success.

The AddonSoftware Partner Program is a uniquely modern approach to structuring a software partnership program. After analyzing the market and noting the advantages and disadvantages of many current programs, we infused open source development concepts with a traditional OEM partnership to create what we believe is the most balanced, dynamic, and open offering available anywhere.

What is Commercial Open Source?

Incorporated into the AddonSoftware Partner Program is a unique component referred to as *Commercial Open Source*, the ability to contribute to the development of the solution itself and in return gain rewards. Participation is optional and the degree of participation is up to each partner.

At its core, *Commercial Open Source* is open source in that partners are contributing directed development to the improvement of the product by adding their own experience with vertical solutions and customer requirements. The commercial twist we added to the open source model is simply that partners receive product credits, up to 100% on new purchases; and a significantly reduced monthly partner fee for their directed 'sweat-equity.'

How Does it Work?

The Partner Program offers three-levels; Authorized, Elite, and Premier. The entry level is the Authorized Partner, which has no sales volume requirements or membership fees, yet still offers a partner discount and no-cost product training. Removing the financial barriers to join and providing you with everything to get started makes this a low-risk opportunity with great potential rewards to participants. Authorized Partners can optionally participate in *Commercial Open Source*, to earn product credits for their development efforts, by paying the same membership fee as our Elite and Premier Partners.

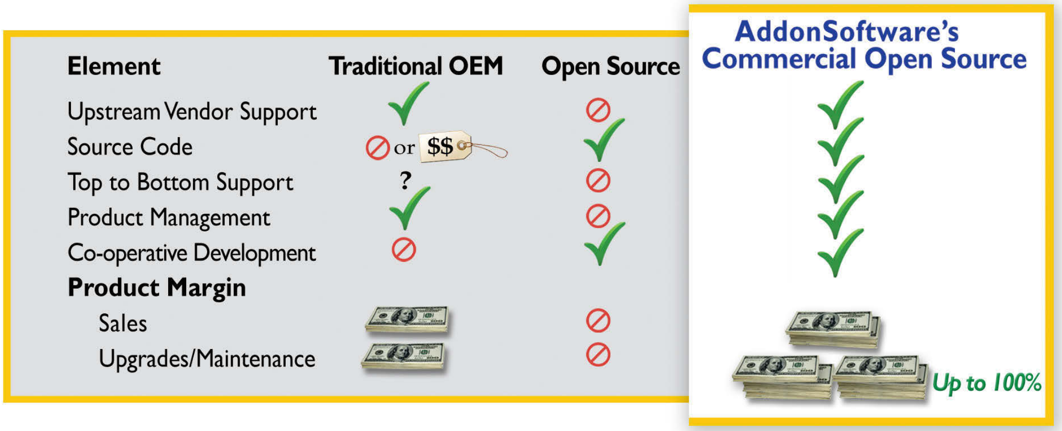
Elite and Premier Partners enjoy higher product discounts based upon product sales volume and optionally may participate in *Commercial Open Source* at no additional charge. Participation also reduces membership fees. Additional no-charge benefits such as passes to technical conference and technical training are included for the Elite and Premier Partners and Authorized Partners who elect to participate in *Commercial Open Source*.

How to Get Started

Sign up for 4½ days of no-cost product training conveniently delivered to you at your desktop by our training team. Interested in development and customization of the solution? Attend an additional 4 days of development tools training, again at no cost. Rather than scheduling all training days continuously, we stagger them among non-training days to give attendees time to practice their new skills and to catch up on work demands, as needed.

How it Compares

Blending the strength of the traditional OEM relationship with features from the open source world creates AddonSoftware's unique *Commercial Open Source* component. Have a look at **Figure 1** to see how this compares with other programs.



Element	Traditional OEM	Open Source	AddonSoftware's Commercial Open Source
Upstream Vendor Support	✓	⊘	✓
Source Code	⊘ or $$	✓	✓
Top to Bottom Support	?	⊘	✓
Product Management	✓	⊘	✓
Co-operative Development	⊘	✓	✓
Product Margin			
Sales	💵	⊘	💵
Upgrades/Maintenance	💵	⊘	💵 Up to 100%

Figure 1. The benefits of the *Commercial Open Source* component

Profitability should be a primary consideration when reviewing a reseller partner opportunity. Can I make money? We preserved the structure of an established market price for the solution; something that open source cannot deliver since there is no selling price or street value for open source products. The competitive partner discounts on product sales, upgrades, and maintenance can be further extended should you decide to leverage your existing expertise via *Commercial Open Source*. You are in a position to make code contributions to the solution, empowered to set your own selling margins. Up to 100% of your new product sales can fall directly to your bottom line!

Summary

Blending the best of both worlds – the traditional OEM VAR channel with the open source model – results in a hybrid AddonSoftware Partner Program that delivers benefits to everyone...end users, developers, and suppliers. Find out more about your opportunities in this uniquely modern program, contact me at pyeomans@addonsoftware.com. ■

- Visit addonsoftware.com
- Review the AddonSoftware Partnership Program Overview at links.addonsoftware.com/partneroverview
- Cick the image below for an interactive peek at this ERP opportunity

We've Got Mail, so You've Got Mail!

Email has become such a huge part of our daily lives, so much so, that we depend on it constantly. From emailing colleagues the latest weekly report to sharing articles and other information with friends, we email throughout the week and weekend on a variety of devices. Wouldn't it be great if your app had tightly integrated email support, complete with contact management and attachment capabilities? That's what BASIS thought too, so read on and discover this new reality with the addition of two new Email Utility classes and how BBJasper and Barista® can now access them.

EmailDialog Class

The new EmailDialog Class is a vital new addition to the Email utility that allows developers to access a built-in BBj® email program to send a message with an optional attachment. Because this BASIS program's source is native BBj code, the user can send emails from within any BBj program – GUI or BUI. Launching this new feature displays the simple and intuitive window shown in **Figure 1**.

EmailDialog uses the standard set of controls found in any typical email program – "To," "Subject," "Message" – and provides the ability to add an attachment to the message. There are also Cc and Bcc options (**Figure 2**) available through the dropdown menu located in the "To" menu button.

Configuring the EmailDialog is very simple. Just click the [Settings...] button at the bottom of the main window (**Figure 1**) and complete the information in the Email Settings dialog (**Figure 3**); "From Email"

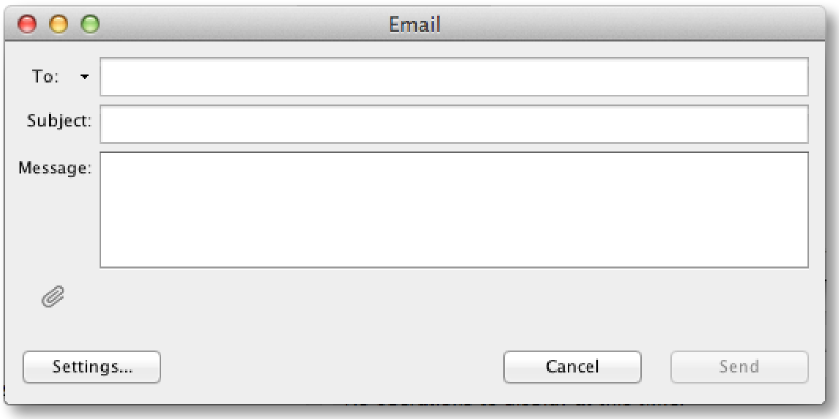


Figure 1. The EmailDialog user interface

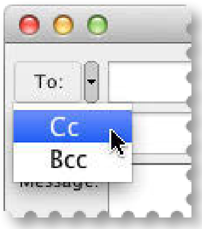
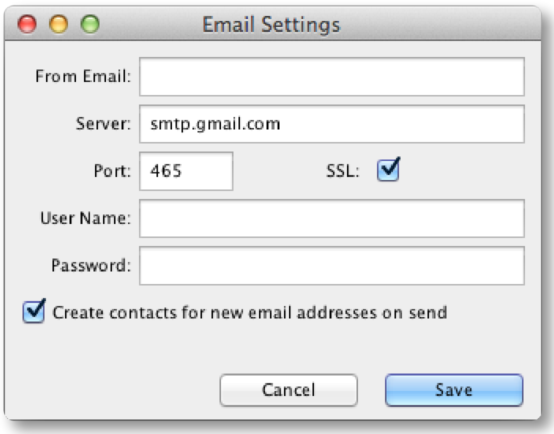


Figure 2. The Cc and Bcc options located in the To: dropdown



Email Settings

From Email:

Server: smtp.gmail.com

Port: 465 SSL: ✓

User Name:

Password:

✓ Create contacts for new email addresses on send

Cancel Save

Figure 3. Configuring the utility via the Email Settings dialog

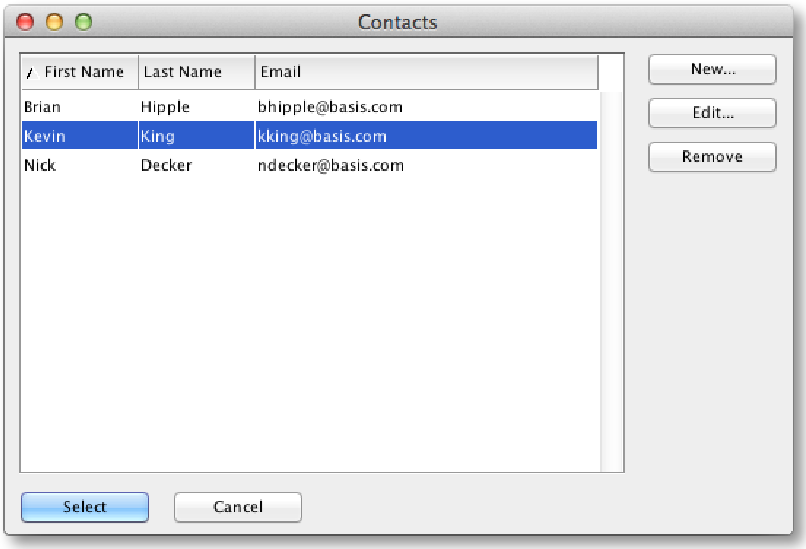
By Amer Child
Digital Communications/ Web Developer

address, mail "Server" and "Port," "User Name," and "Password." By default, the mail server uses Google and SSL for secure communication over the Internet, but those parameters are configurable to match your needs.

EmailContact Class

With any email program, it is desirable to have a contacts list. The second class that BASIS added to the Email Utility is EmailContact, activated by marking the checkbox "Create contacts for new email addresses on send." Users can simply access their own contact list with a click on the [To:] button, and their list of available contacts displays in a new window, as shown in **Figure 4**.

The built-in contact list stores the information on the client system for each user to access and maintain individually. To send an email to a contact in the list, the user clicks the name, or multiple names with [Ctrl]+click, and then clicks [Select]. From inside this list, users can also manually add new contacts, or edit or remove an existing contact by clicking the entry and selecting [Edit...] or [Remove].



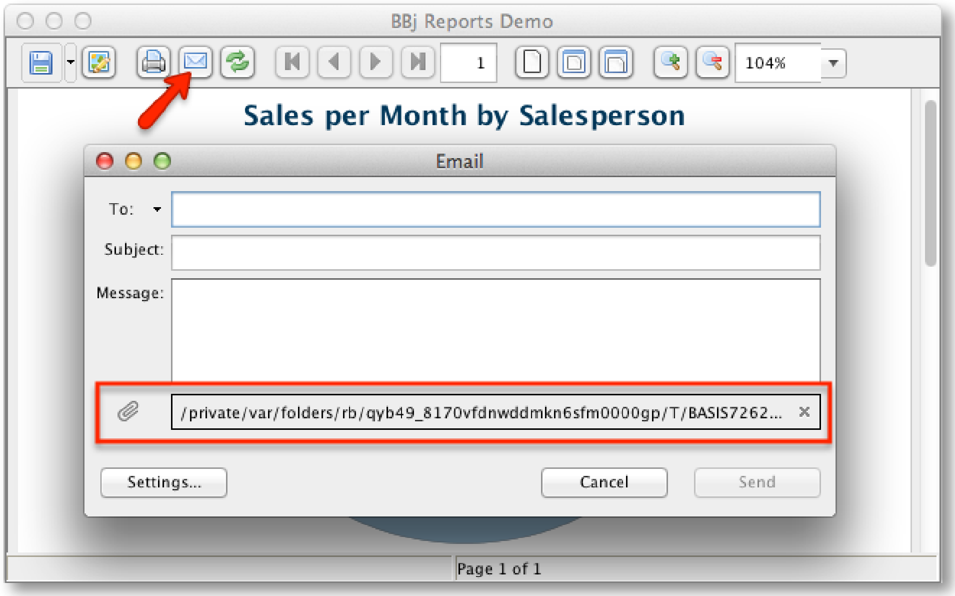
First Name	Last Name	Email
Brian	Hipple	bhipple@basis.com
Kevin	King	kking@basis.com
Nick	Decker	ndecker@basis.com

Figure 4. A sample contacts list

BBJasper, You've Got Mail!

With the addition of these new classes, BASIS has given BBJasper access to the Email utility, greatly enhancing reporting output capabilities. See the email button that now appears on the menu bar (**Figure 5**) of the BBJasperReport viewer window, which launches the EmailDialog. Notice that BBJasper automatically adds a link to the viewed report in the attachment field.

Similar to the BBJasperViewer toolbuttons, developers can programmatically disable, hide, or even override the email button to call their own custom code. The email functionality is easily added to your applications as well, paving the way for a one-click email attachment.



/private/var/folders/rb/qyb49_8170vfdnwddmkn6sfm0000gp/T/BASIS7262...

Figure 5. Attaching a report to an email in BBJasper

Barista, You've Got Mail!

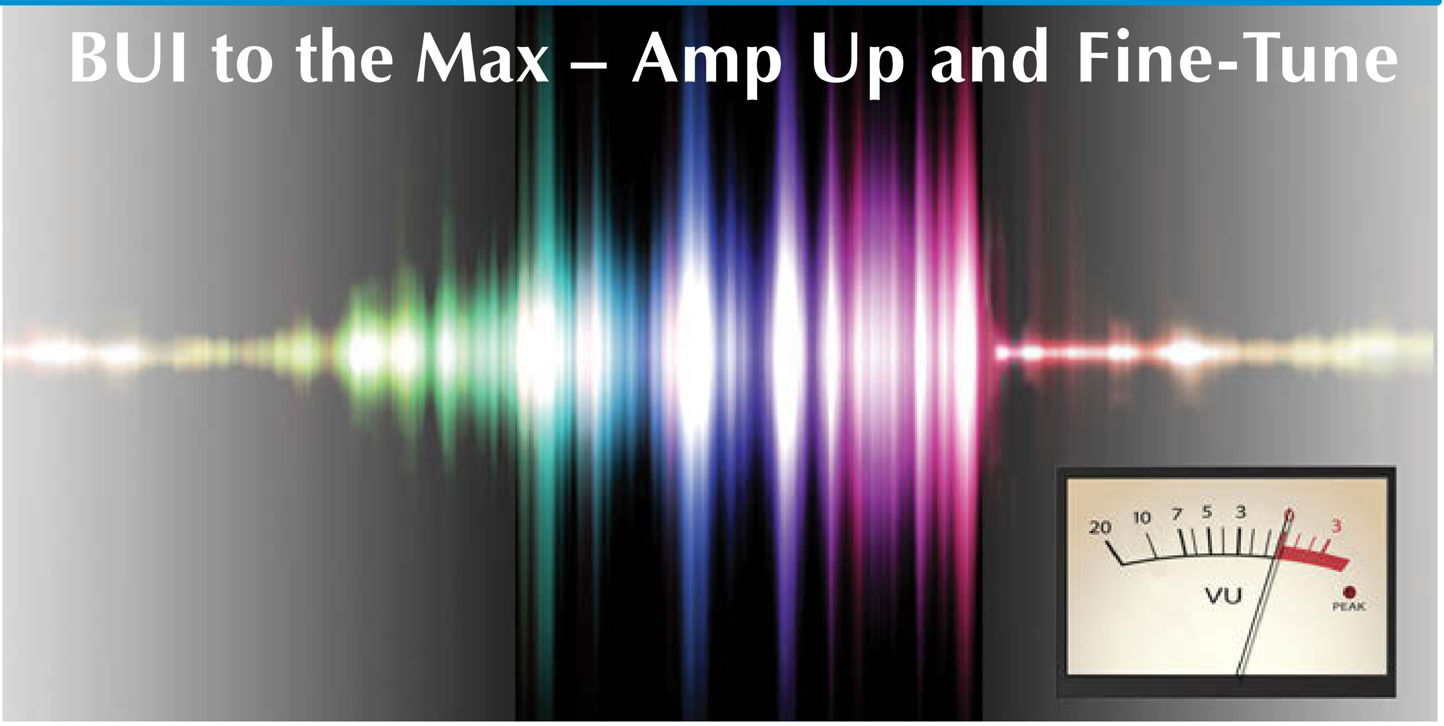
When viewing a BBJasper report from within Barista, clicking the email button launches Barista's own Document Queue Information form rather than the EmailDialog form, so the user can add the report to the Barista Document Queue. By implementing the feature in this way, BASIS allows for consistency between your custom application, DocOut, and/or JasperReports, giving a consistent and more intuitive user experience with any of the components you use in your application.

Summary

With new BASIS' Email utility classes – EmailDialog and EmailContact – you can now easily add value to any BBj application with a new email solution. Even sending attachments within your application is as simple as the click of a button. Built-in contact management allows users to easily manage and maintain their own contacts for quick distribution of important information. The EmailDialog class is another great addition to the BASIS toolset, making the development of your application even easier and more user-friendly. And the price is right – no cost! Add value to your business apps and tell your users, "You've Got Email!" ■

Have a look at the online help for the full no-cost Email utility at links.basis.com/emaildocs

BUI to the Max – Amp Up and Fine-Tune



Most Visual PRO/5® or BBj® GUI applications are inherently BUI applications. In the minute it takes to configure an app in your BUI app server, you're good to go. But a working BUI app is just the first step. This article covers some easy things you can do to maximize the performance and fine-tune the appearance to make it more like apps that users expect of a web page and less like a transplanted desktop application.

Getting Started

BASIS has already taken the first step: BBj 13.10 previews a completely redesigned BUI theme for BBj 14. This new theme includes hundreds of aesthetic and functional tweaks to all BUI controls. On touch devices, dialogues like MSGBOX() and FILEOPEN() are better centered, and some controls have been modified to work better. For example, spinner buttons are oriented horizontally to provide larger touch targets. Even if you do nothing at all, your BUI apps already look better than ever before! Find out more details about the new defaults in *Default CSS Gets a Makeover* at links.basis.com/13css.

By Jim Douglas
Software Developer

Performance

We first talked about bandwidth and latency issues with distributed applications in *The Lessons of BASIS b-Commerce* (links.basis.com/00bcomm). Now, 13 years later, these issues are still a major consideration with client/server applications. We touched on this subject in *The Anatomy of a Web App Makeover* (links.basis.com/12webapp), which looks closely at the BUI version of the BBj download page. When we design a distributed application, the client and server work together to manipulate data, perform calculations, and present information to the user. The communication between the client and server can be broken down into three broad categories.

Category 1: Server to Client – includes methods like **`BBjEditBox::setText`**. The BBj application running on the server sends data to the client without waiting for a response. BBj automatically optimizes batches of operations in this category to improve performance.

Category 2: Client to Server – includes event traffic like the ON_BUTTON_PUSH event. For the most part, the client sends events to the server without waiting for any sort of response. The event object typically contains additional parameters relating to the event such as, for example, the width/height of a resized window, or the x,y location of a moved window. Because this information is delivered to the server as part of the act of delivering the event, it is immediately available to the program as soon as the event is received.

Category 3: Server to Client to Server (round trip) – includes methods in which the application queries the client for some dynamic information that cannot be cached on the server. Methods like **`BBjEditBox::getText`** and **`BBjCheckBox::isSelected`** fall into this category. When a BBj application executes a line like **`name$ = editbox!.getText()`**, it comes to a complete standstill while the server sends that request to the client, then waits for the client to send back the response. As a rough rule of thumb, we usually assume that each round trip like this takes at least 100 ms (1/10th of a second), but it can easily be double that time, especially on a mobile device. And it gets worse: If the application sends several messages to the client that don't require a response (Category 1), followed by a single message that requires a response (Category 3), the application must wait for the client to process all pending messages, then respond to the final message that requires a response. In some cases, this can add hundreds of milliseconds to the delay. If that round trip hadn't been introduced, the client would have been able to continue working through processing those backlogged messages while the application running on the server moved on to new work.

We can significantly improve application responsiveness by avoiding category 3 (synchronous round trips) as much as possible. For example, we can change the ON_LIST_CLICK event handler for a BBjListBox from this:

```
on_list_click:
    event! = bbjapi().getSysGui().getLastEvent()
    listbox! = event!.getControl()
    index = listbox!.getSelectedIndex(); rem ' goes to the client
    rem -- do something here ---
return
```

to this:

```
on_list_click:
    event! = bbjapi().getSysGui().getLastEvent()
    index = event!.getSelectedIndex(); rem ' gets value from event
    rem -- do something here ---
return
```

In the first version, BBjListBox::getSelectedIndex forces a round trip back to the client, typically introducing a delay of 100 milliseconds or more. In the second version, BBjListClickEvent::getSelectedIndex retrieves the value that was delivered to the server as part of the original event.

To see the effect of eliminating round trips, run the BBjFormValidation demo (**Figure 1**) in the BBx BUI Showcase at links.basis.com/buidemos.

The program shows 14 controls of various types, with a navigator to browse through 100 records containing randomly generated data. The UI is smooth and fast; clicking the navigator buttons brings up records as fast as you can click. The two big buttons at the bottom of the form implement two different ways to retrieve the current contents of the form. The "Form Validation" button fires a BBjFormValidationEvent, which captures the current values of all user-changeable controls on the window. The "getText" button fires a BBjButtonPushEvent. In response to that event, the application queries each of the 14 controls for their current value. The difference in speed is dramatic.

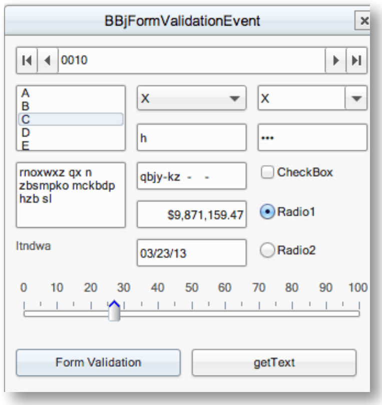
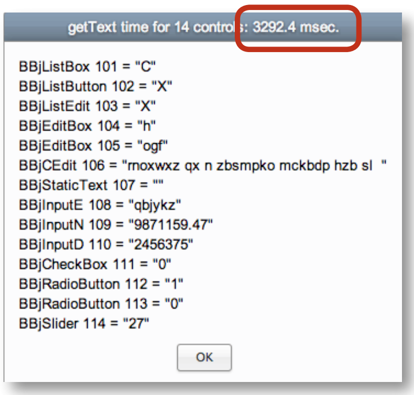


Figure 1. The BBjFormValidation demo

The getText version takes at least 1.5 to 2 seconds, but can take several seconds depending on the state of your Internet connection and the latency to the server machine. When we ran the demo here at BASIS, the getText version took more than 3 seconds, as shown in the title bar of the resulting dialog shown in **Figure 2**.



```
getText time for 14 controls: 3292.4 msec.

BBjListBox 101 = "C"
BBjListButton 102 = "X"
BBjListEdit 103 = "X"
BBjEditBox 104 = "h"
BBjEditBox 105 = "ogf"
BBjCEdit 106 = "rnoxwxz qx n zbsmpko mckbdp hzb sl  "
BBjStaticText 107 = ""
BBjInputE 108 = "qbjykz"
BBjInputN 109 = "9871159.47"
BBjInputD 110 = "2456375"
BBjCheckBox 111 = "0"
BBjRadioButton 112 = "1"
BBjRadioButton 113 = "0"
BBjSlider 114 = "27"
```

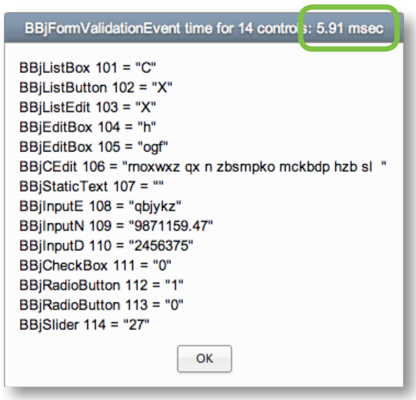
Figure 2. The getText results taking 3,292.4 milliseconds

The Form Validation version (**Figure 3**) takes effectively no time. It has all the data it needs as soon as the event hits the server, so it reports all of the same information 557 times faster, in a scant 5.91 milliseconds (0.00591 seconds).

Touch Click

If you skim the source code (links.basis.com/formvalidation-code) for that form validation sample program, you might wonder about this strange line:

```
fast_touch_click$ = stbl("!OPTIONS","FAST_TOUCH_CLICK=TRUE")
```



```
BBjFormValidationEvent time for 14 controls: 5.91 msec

BBjListBox 101 = "C"
BBjListButton 102 = "X"
BBjListEdit 103 = "X"
BBjEditBox 104 = "h"
BBjEditBox 105 = "ogf"
BBjCEdit 106 = "rnoxwxz qx n zbsmpko mckbdp hzb sl  "
BBjStaticText 107 = ""
BBjInputE 108 = "qbjykz"
BBjInputN 109 = "9871159.47"
BBjInputD 110 = "2456375"
BBjCheckBox 111 = "0"
BBjRadioButton 112 = "1"
BBjRadioButton 113 = "0"
BBjSlider 114 = "27"
```

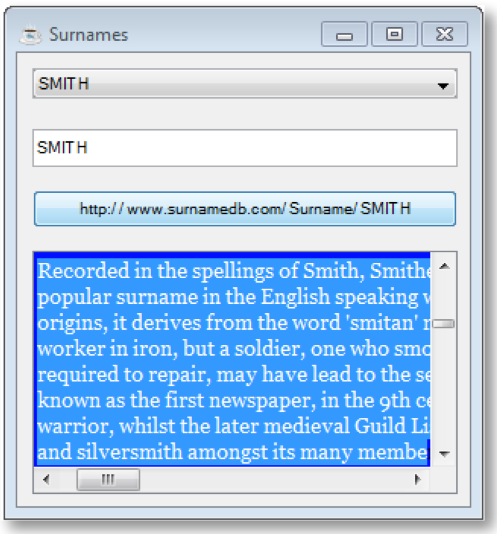
Figure 3. The Form Validation results taking only 5.91 milliseconds (557 times faster)

In touch-oriented browsers, the user can double-tap anywhere on the screen to zoom in and out. To allow for the possibility that any given tap might be the start of a double-tap gesture, mobile browsers wait for about 300 milliseconds (0.3 seconds) before reporting that a button was clicked. Most of the time, this isn't a problem. The slight delay is a tradeoff for the added usability of being able to double-tap anywhere to zoom. But there are times when you want the user to be able to rapidly click a button and have the program respond to the click immediately.

In the default mobile browser configuration, clicking a button twice within less than 300 ms isn't interpreted as two clicks; it's interpreted as a double-tap gesture to zoom the window. When specifying the FAST_TOUCH_CLICK option, button controls (and the buttons in navigator controls) report click events immediately, eliminating that 300 ms delay. (Of course, this has the effect of disabling the standard double-tap-to-zoom behavior when the user taps directly on a button, which is why it's a developer-configurable option, as opposed to standard BUI behavior.) When selecting this option, the user is still able to double-tap to zoom anywhere else on the form, just not directly on a button.

From GUI to BUI

Let's take a small GUI app, run it in BUI, and see what we might want to do to optimize it for the browser environment. For demonstration purposes, we'll use a trivially simple application, one that prompts for a last name, or surname, and displays information about it. **Figure 4** is the original version running in GUI.

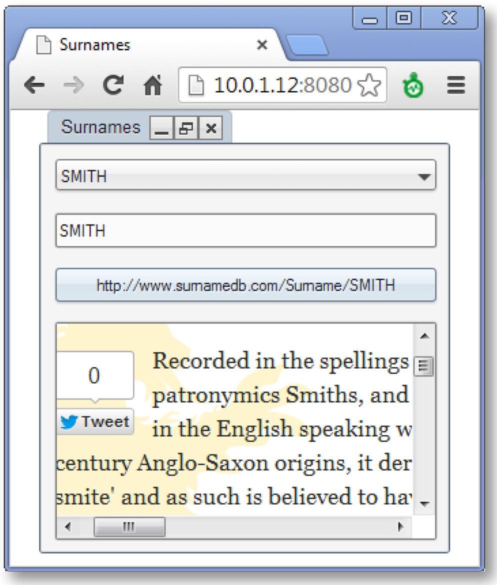


Recorded in the spellings of Smith, Smithe
popular surname in the English speaking w
origins, it derives from the word 'smitan' r
worker in iron, but a soldier, one who smo
required to repair, may have lead to the se
known as the first newspaper, in the 9th ce
warrior, whilst the later medieval Guild Li
and silversmith amongst its many membe

Figure 4. Surname sample running in GUI

Pretty basic stuff – a BBjWindow with a BBjListButton to select a surname from a list, a BBjEditBox to type a name that might not appear on that list, a BBjButton that the user can click to query the currently selected name (identified as a URL on the button), and a BBjHtmlView to show information about that name.

Figure 5 shows the same program running in BUI.



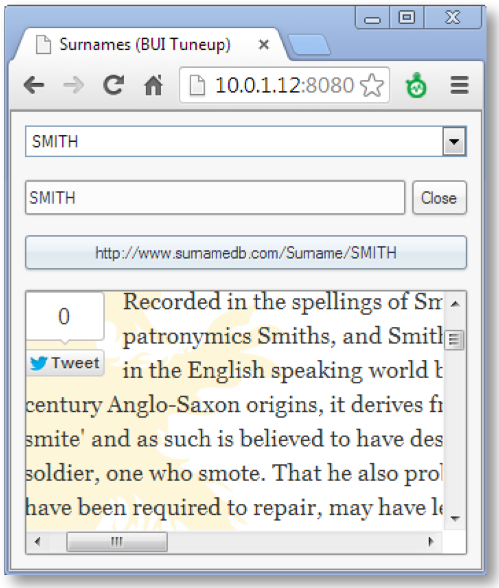
Recorded in the spellings
patronymics Smiths, and
in the English speaking w
century Anglo-Saxon origins, it der
smite' and as such is believed to ha

Figure 5. Surname sample running in BUI (Chrome)

That title bar looks a bit out of place in a web page, and it would be nice if we used all of the available space. So let's try a few tweaks:

- ☑ Create the window with no title bar; optionally add a separate Close button.
- ☑ Maximize the window to use the full browser client area.
- ☑ Use the NATIVE_BROWSER_LIST version of the listbutton; it can provide a better user experience on mobile devices.
- ☑ Size and position all of the controls based on the available space.
- ☑ Add a resize handler to resize the controls when the user resizes the browser (or changes the orientation of a mobile browser).

Figure 6 looks much more like a standard web page, and less like a desktop application running in a browser.

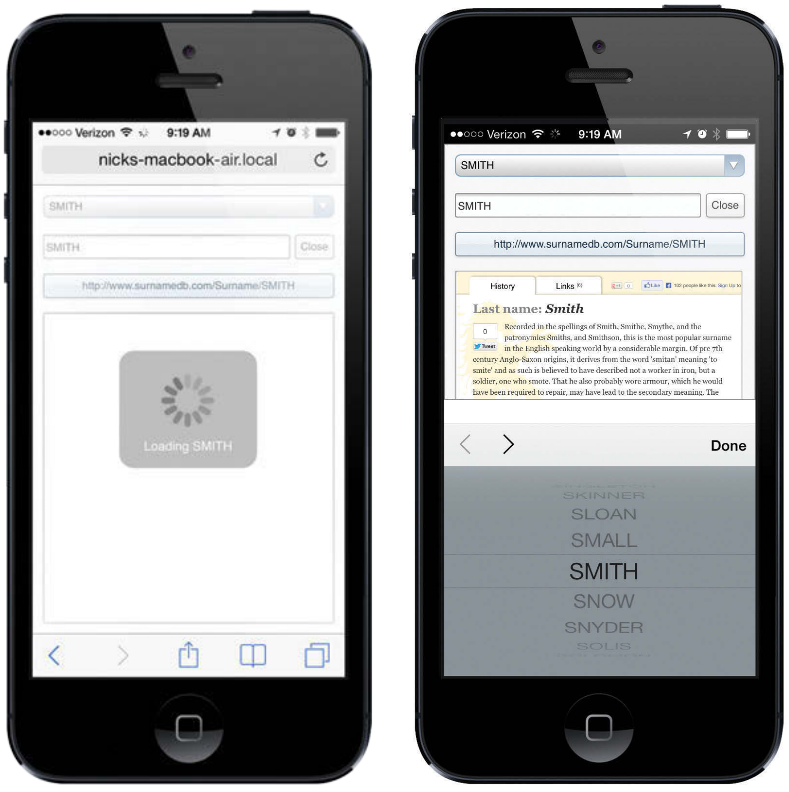


Recorded in the spellings of Sm
patronymics Smiths, and Smith
in the English speaking world b
century Anglo-Saxon origins, it derives fr
smite' and as such is believed to have des
soldier, one who smote. That he also prol
have been required to repair, may have le

Figure 6. Surname sample in BUI with some simple modifications

Now let's try it on a mobile device. This shows us one more opportunity for improvement. In desktop applications (both GUI and BUI), we can set the wait cursor to indicate that the application is temporarily busy. Mobile devices don't have cursors, so BUI offers another mechanism, the BBjBusyIndicator, to indicate that the application is busy. We can show the busy indicator when we set the wait cursor, then remove it when we reset the cursor.

The iPhone on the left in **Figure 7** shows how the busy indicator looks on a mobile device. Notice that the browser's top URL Bar and bottom Button Bar consume an appreciable amount of the already-limited screen real estate. On most mobile devices, you can add the application to the home screen with a few taps, which results in making the BUI app look and feel similar to a native application. The BUI app will then have its own icon on the home screen, will run in "standalone" mode instead of in the browser, and will take up the entire screen space except for the top



Last name: *Smith*

Recorded in the spellings of Smith, Smithe, Smythe, and the patronymics Smiths, and Smithson, this is the most popular surname in the English speaking world by a considerable margin. Of pre 7th century Anglo-Saxon origins, it derives from the word 'smitan' meaning 'to smite' and as such is believed to have described not a worker in iron, but a soldier, one who smote. That he also probably wore armour, which he would have been required to repair, may have lead to the secondary meaning. The

Figure 7. Surname sample on a mobile device in the browser (left) and in standalone mode (right)

status bar. The iPhone on the right in **Figure 7** shows the Surname sample running in standalone mode from the home screen. Because the NATIVE_BROWSER_LIST option was set, you choose a surname from the BBjListButton by selecting a name from the native iOS picker control in place of the BBjListButton's dropdown list.

The constrained viewport on mobile devices presents another challenge compared to desktop devices. We should take this into account and limit our display to just a few important items when designing BUI apps for deployment to mobile devices.

We can also take advantage of another new BUI feature: custom end actions. Custom end actions allow the developer or system administrator to define what happens when the user exits a BUI app.

```
click:
   htmlview!.setUrl(url$+name$)
   window!.setCursor(3)
   if busy!<> null() then
      busy!.setText("Loading " + name$)
      busy!.setVisible(1)
      bui!.setEndAction(bui!.urlAction(url$+name$))
   endif
return
```

Figure 8. Sample code to set the custom BUI end action at runtime

When the program loads the selected surname, it can tell BUI to chain to the URL for that name upon termination of the BUI app in **Figure 8.** This code lets the user load a preview of the name information into the htmlview on the BUI app page, then go to a full-page view of that same information when closing the BUI app page. This feature has many potential uses, including building full application menuing systems in BUI.

Summary

BBj's browser user interface has been around for several years. Over time it has matured, and now runs faster, looks better, and provides the application developer with several new configuration and customization options. The addition of form-level validation and more payload data in various events significantly reduces client-server round trips, speeding up the execution of remotely-deployed BUI apps and making them feel more like local applications. BASIS has also added several improvements to streamline and improve BUI apps running on mobile platforms, including native message boxes and list buttons, fast touch click detection, a built-in BBjBusyIndicator, and the ability to easily create a fullscreen app without a titlebar. Lastly, custom end actions give developers the power to determine where their BUI app should take the user after it has finished executing. BASIS' browser user interface is an exciting technology that continues to expand and improve. If you've not begun to take advantage of it yet, now is the time for you to amp up and fine-tune your BUI application. ■ links.basis.com/**13code**

- Refer to *Default CSS Gets a Makeover* at links.basis.com/13css
- Read about the BBjBusyIndicator and custom end actions in *Automate BUI Deployment With the API* at links.basis.com/13buiapi
- See more mobile-optimized sample applications in the entire *BBj BUI Showcase* at links.basis.com/buidemos
- How to add a BUI app to your mobile device's home screen:
  - iOS: links.basis.com/iphone-addicons
  - Android: links.basis.com/android-addbookmarks

Biografie - Eigenschaften - Abend, Johannes

Vorlieben: Arbeiten, [✓ Gartenarbeit], [✓ Einkaufen], Kochen, Backen, Aufräumen, Fernsehen, [✓ Lesen], Spazieren gehen, [✓ Reisen], Singen, Musik hören

Sonstiges: liest gern die Freie Presse, sieht gern fern, geht gern einkaufen

Verhaltensweisen: sportlich, [✓ gesundheitsbewusst], modebewusst, [✓ körperbewusst], geschlechtsbewusst, mutig, [✓ ruhig], mitteilsam, laut, fröhlich, ängstlich

Sonstiges:

Lieblingsthemen: Fußball (Schalke 04)

Job Details - BASIS International

Gregory Baldrake (0105532)

320 xx DEPT 104 5th YR APPR

A5 Apprentice (Fifth Year)

001 Various FWOs

Day Shift — FWO9001

Standard	Overtime	Double Time
8.0	2.5	0.0

How Business Apps Go Mobile

Mobile apps have generated a completely new market. According to an InMobi study, "*50% of the average global mobile web users now use mobile as either their primary or exclusive means of going online*" (links.basis.com/fibnd). ISVs with an exclusively business-oriented product portfolio need to offer mobile apps in order to stay competitive. This article provides an overview about the possible approaches and their suitability for business applications.

Mobile Applications are a Must-Have

The market for mobile applications is booming while, in terms of smartphone operating systems, there are not many players left. According to the International Data Corporation (IDC), "*Android and iOS powered 85% of all smartphones shipped in the second quarter of 2012*" (links.basis.com/ahdas).

Seen from a programmer's point of view, though, the mobile apps market still looks pretty unclear. There is no "one size fits all" way to write native apps that would work on all OS platforms. Too different are the specific UI concepts. So, at first sight it seems you would either have to write a separate native app for each platform you intend to cover, or make use of the fact that all mobile devices have Internet access and are equipped with web browsers. But web apps have their downsides as well, which we'll discuss later.

To make things worse, most ISVs do not start with a blank slate; they already have their desktop applications to maintain. In most cases, they just want their mobile app to offer an add-on functionality of their desktop application for specific user groups, such as salesmen, technicians, or management who cannot easily access the company's IT system when they are mobile. Developing several parallel mobile apps that would require working in different development environments while at the same time having to assign resources to the maintenance of their main application, is not a feasible option.

Tools for Cross-Platform Development

With over 100 tools on the market meant to enable or simplify cross-platform development, developers can choose among three basic ways to avoid writing multiple native apps - cross-compiled apps, hybrid apps or web apps (**Figure 1**). We will discuss these ways and, additionally, will see how BASIS' Browser User Interface (BUI) technology fits into the overall picture.

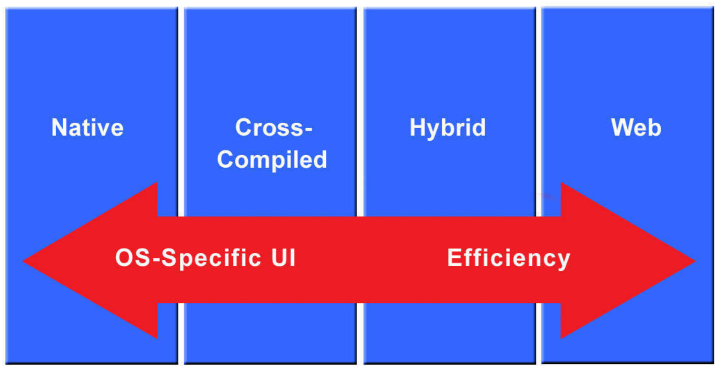


Figure 1. There are four basic ways to write apps for multiple mobile device platforms.

By Patrick Schnur
European Marketing/PR

Native Apps...Snazzy to Run, Cumbersome to Make

Developers design native apps for a given operating system in the respective programming language – Java for Android and BlackBerry, Objective C for iOS, .Net for Windows, and so on. By programming native apps, you make sure that your application can make full use of all hardware resources of the device, and that all interfaces work together uniformly. Therefore, they are best for computation-intensive applications, and are preferred for visualizing 3-D graphics and for games. **Figure 2** illustrates the typical structure of a native app.

The obvious downside are the costs for developing and maintaining several applications in different programming languages meaning a steep learning curve, plus implementing different platform-specific methodologies and compliance with rules and regulations to deploy the apps.

Cross-Compiling Apps the Best of all Worlds?

The idea behind cross-compiled apps is to write the source code just once and then compile it via code generators to the various platforms. What you get are in principle native apps, but based on the lowest common denominator in terms of functions and the UI. The downside is, again, a pretty steep learning curve and no saving on the deployment hurdles. Cross compiling apps with a code generator as shown in **Figure 3** gives you the lowest common denominator of all native UIs.

Hybrid Apps Mind the Gap!

Hybrid apps try to combine the advantages of native apps and web apps. Technically, hybrid apps are HTML5- and JavaScript-based web apps connected by containers written in the respective native languages. Hybrid apps can access at least some of the hardware resources of smartphones and tablets, without the need to write completely separate source code for each platform. Such functions are directly using the camera, the contact database, media information, device features and the device's internal database. The native containers needed to achieve this are bundled in a framework such as Phonegap. Hybrid apps generated with frameworks like "Phonegap" allow mobile apps to access some of the hardware resources of smartphones and tablets as illustrated in **Figure 4**.

Web Applications

Web apps, obviously, are applications which uses a web browser as the client. Since nearly all commonly used mobile devices are equipped with such browsers, web apps are a convenient way to reach potentially all target groups with minimum development effort, and none of the deployment headaches.

But web apps written for the desktop often sport a completely different UI than would be expected for smartphones or tablets, and the user interface is important for user acceptance. Therefore, web apps usually need tweaking to make them more resemble the look-and-feel of native apps which consumers are used to. It is not possible to mimic the native UIs completely, as they

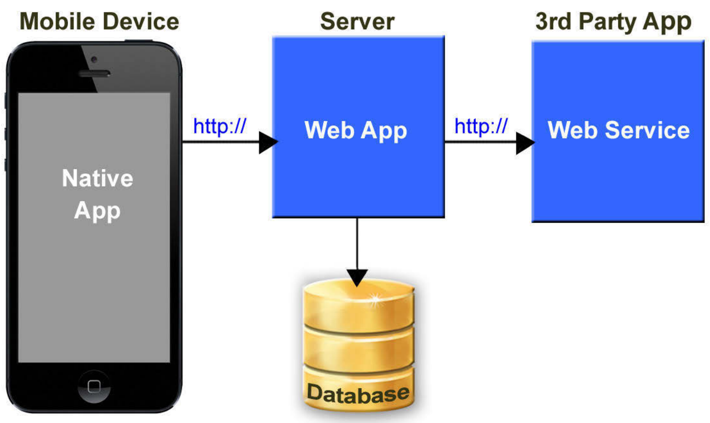


Figure 2. Typical architecture of a native application

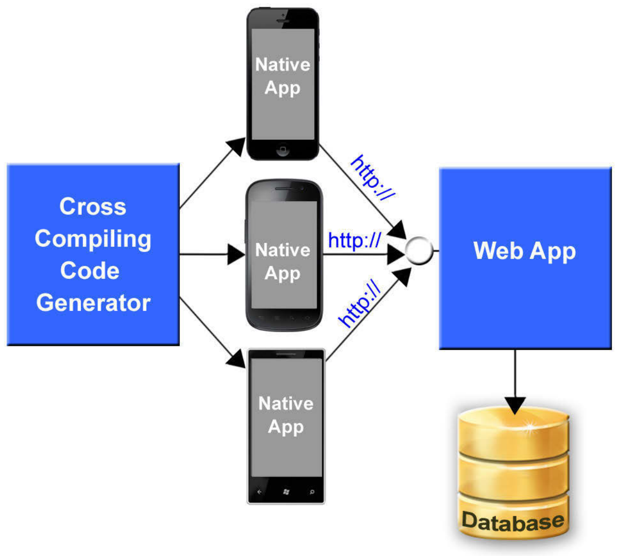


Figure 3. Typical architecture of a code generator

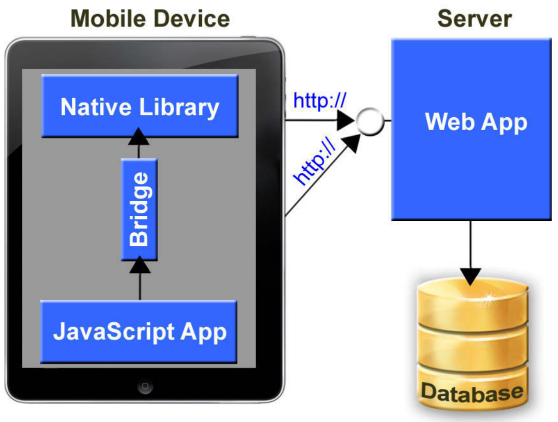


Figure 4. Typical architecture of a hybrid app

differ from each other significantly, but it generally is possible to adapt to the needs of small screens and gesture-controlled touchscreens. See **Figure 5.**

Where Does BUI fit in?

With the browser user interface, or BUI for short, BASIS offers a fifth approach to developing cross-platform mobile apps. BUI s implemented with GWT (www.gwtproject.org), which cross-compiles Java to JavaScript and enables us to create a browser-based BBj client using standard HTML, CSS, and JavaScript.

The Heimbas (www.heimbas.de) BUI application for care management in nursing homes runs on iPads and all other tablets that are equipped with a JavaScript-enabled browser. During the doctor's visit, all relevant patient information is accessible at the tap of a finger, right at the bedside. **Figure 6** and **Figure 7** shows examples of the app running on iPad, examples of patient care and quality management made easy.

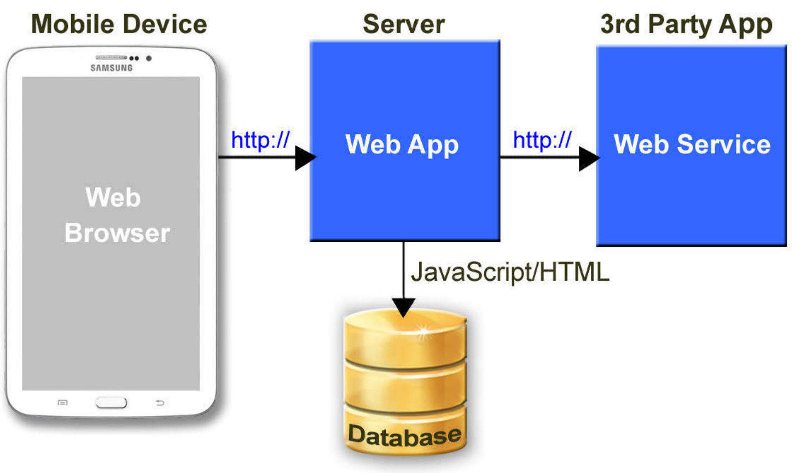
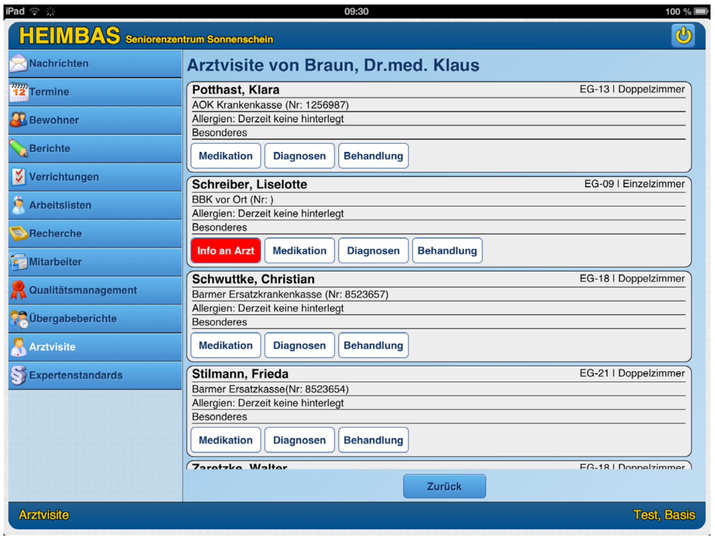


Figure 5. Typical architecture of a server-side web app



iPad 09:30 100 %

HEIMBAS Seniorenzentrum Sonnenschein

Nachrichten
Termine
Bewohner
Berichte
Verrichtungen
Arbeitslisten
Recherche
Mitarbeiter
Qualitätsmanagement
Übergabeberichte
Arztvisite
Expertenstandards

Arztvisite von Braun, Dr.med. Klaus

Potthast, Klara — EG-13 I Doppelzimmer
AOK Krankenkasse (Nr: 1256987)
Allergien: Derzeit keine hinterlegt
Besonderes
Medikation | Diagnosen | Behandlung

Schreiber, Liselotte — EG-09 I Einzelzimmer
BBK vor Ort (Nr:)
Allergien: Derzeit keine hinterlegt
Besonderes
Info an Arzt | Medikation | Diagnosen | Behandlung

Schwuttke, Christian — EG-18 I Doppelzimmer
Barmer Ersatzkrankenkasse (Nr: 8523657)
Allergien: Derzeit keine hinterlegt
Besonderes
Medikation | Diagnosen | Behandlung

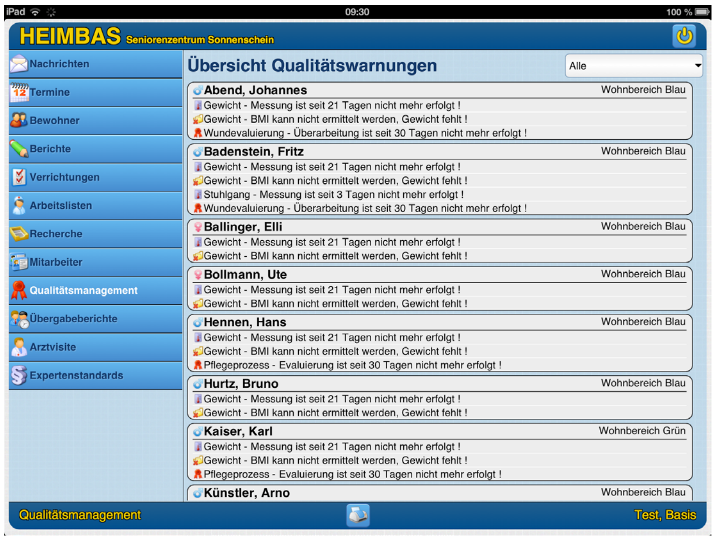
Stilmann, Frieda — EG-21 I Doppelzimmer
Barmer Ersatzkasse(Nr: 8523654)
Allergien: Derzeit keine hinterlegt
Besonderes
Medikation | Diagnosen | Behandlung

Zaretzke, Walter — EG-18 I Doppelzimmer

Zurück

Arztvisite — Test, Basis

Figure 6. Heimbas Care Management running on iPad



iPad 09:30 100 %

HEIMBAS Seniorenzentrum Sonnenschein

Nachrichten
Termine
Bewohner
Berichte
Verrichtungen
Arbeitslisten
Recherche
Mitarbeiter
Qualitätsmanagement
Übergabeberichte
Arztvisite
Expertenstandards

Übersicht Qualitätswarnungen — Alle

Abend, Johannes — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !
Wundevaluierung - Überarbeitung ist seit 30 Tagen nicht mehr erfolgt !

Badenstein, Fritz — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !
Stuhlgang - Messung ist seit 3 Tagen nicht mehr erfolgt !
Wundevaluierung - Überarbeitung ist seit 30 Tagen nicht mehr erfolgt !

Ballinger, Elli — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !

Bollmann, Ute — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !

Hennen, Hans — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !
Pflegeprozess - Evaluierung ist seit 30 Tagen nicht mehr erfolgt !

Hurtz, Bruno — Wohnbereich Blau
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !

Kaiser, Karl — Wohnbereich Grün
Gewicht - Messung ist seit 21 Tagen nicht mehr erfolgt !
Gewicht - BMI kann nicht ermittelt werden, Gewicht fehlt !
Pflegeprozess - Evaluierung ist seit 30 Tagen nicht mehr erfolgt !

Künstler, Arno — Wohnbereich Blau

Qualitätsmanagement — Test, Basis

Figure 6. Care Management quality issues that need attention

By generating a JavaScript application from BBj code on the fly, developers don't have the worries of developing and maintaining different applications for desktops, the web and mobile devices, they have an all-in-one solution which will work on any device with a JavaScript-enabled web browser. To adjust the UI to the specific needs of smartphones and tablets, BASIS customers use cascading stylesheets, or CSS, which they do with impressive results.

Emque Consultants (www.emque.com) in New York offers a BUI app for the commercial construction industry to which BASIS mocked up some CSS modifications shown in **Figure 7.**

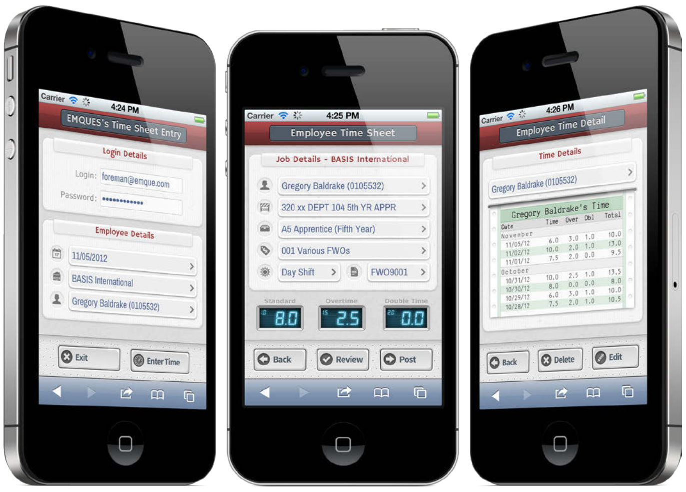


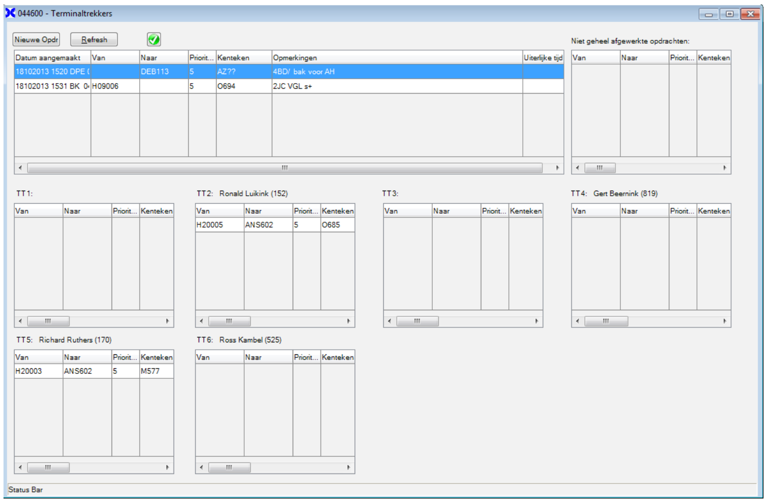
Figure 7. Emque's construction management BUI app with BASIS-supplied CSS

BUI apps can address the GPS sensor of a mobile device with the help of the browser. Logistics service provider Wim Bosman (www.wimbosman.com) used BUI for an onboard tracking system that allows their headquarters in the Netherlands to track their trucks en route across Europe in real-time (**Figure 8**) and assign incoming orders to the best-equipped and best-positioned truck.

While BUI may not be the best solution if you need to handle rich media or plan to create a blockbuster game, BUI is probably the most efficient methodology available in the market for data-driven mobile solutions. BUI is ideal if you need to extend an existing business application to new target groups or if you have to deploy your application to run simultaneously on multiple platforms. Consider how BUI would save you the effort of building a development team capable of applying several programming toolsets and languages. Now you could enjoy the ease of having only one source code to maintain over the life of your application. Furthermore updating the app is a cinch, just publish a new version to your server and all the client devices will get it.

Revenue Sharing

If you are concerned about the licensing costs for the millions of users who will now use your app, contact your BASIS account manager to work out a win-win revenue sharing plan. We promise not to require more than the 30% revenue share that developers currently provide to the major native app store vendors. ☺



Datum aangemaakt	Van	Naar	Priorit...	Kenteken	Opmerkingen	Uiterlijke tijd
18102013 1520 DPE (		DEB113	5	AZ??	4BD/ bak voor AH	
18102013 1531 BK 0	H09006		5	O694	2JC VGL s+	

Niet geheel afgewerkte opdrachten:

Van	Naar	Priorit...	Kenteken

TT1:

Van	Naar	Priorit...	Kenteken

TT2: Ronald Luikink (152)

Van	Naar	Priorit...	Kenteken
H20005	ANS602	5	O685

TT3:

Van	Naar	Priorit...	Kenteken

TT4: Gert Beernink (819)

Van	Naar	Priorit...	Kenteken

TT5: Richard Ruthers (170)

Van	Naar	Priorit...	Kenteken
H20003	ANS602	5	M577

TT6: Ross Kambel (525)

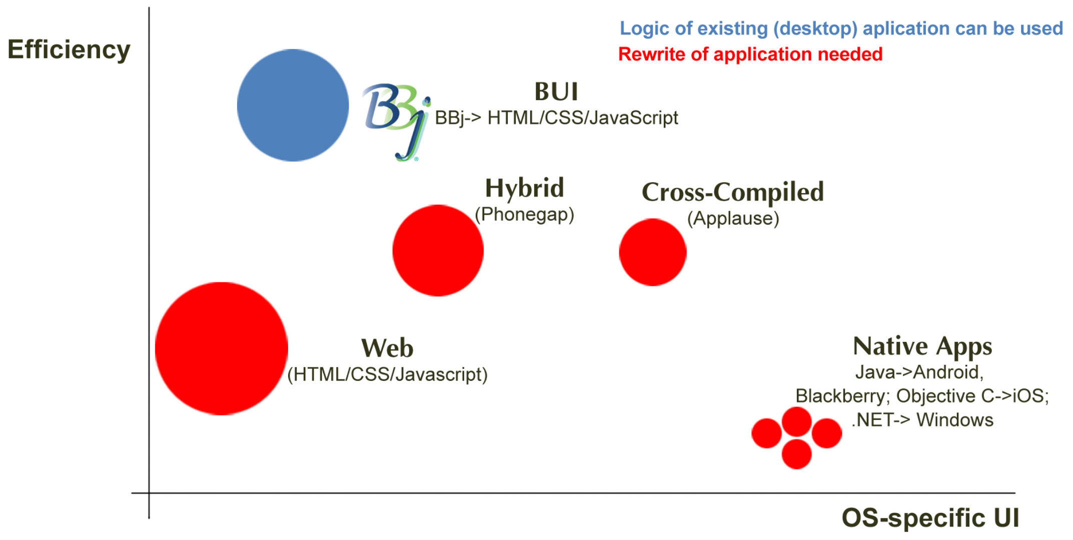
Van	Naar	Priorit...	Kenteken

Status Bar

Figure 8. Wim Bosman's onboard tracking app

Summary

Technologically, there are four basic ways of developing mobile apps. While each has its pros and cons, BUI, a superior web app deployment paradigm, is by far the most cost effective and risk-free way to build mobile data-driven business applications, especially those that are derived from pre-existing desktop applications. And perhaps best of all, there's no app store approval process or multiple deployment and update scenarios to contend with. ■



Efficiency in terms of learning curve, effort of development, deployment and maintenance
OS-specific UI: How well does the user interface mimic the original?

Mobile apps in a nutshell: BUI saves ISVs a lot of time and effort in making mobile apps for various platforms.

Find all the available statistical data about the consumer mobile app market that you could possibly dream of; and in easy to understand graphical format, at links.basis.com/xtvtk

Go to basis.com and CSE search on BUI for a plethora of good reads!

BBj's CUI Gets a Big Performance Boost

Our bodies are extremely efficient at processing the food we eat so that our internal workings run proficiently. If you are like me, when I work around the house like cleaning the gutters and mowing the lawn, I always look for the fastest and easiest way to accomplish those tasks. After all, the faster I can get a task done, the more time I have for more pleasurable activities! I might even look for more creative ways to perform those tasks and take advantage of new technologies. For example, instead of walking to work a few miles away, I might drive a car or take public transportation, or even work "virtually" by way of the Internet.

So why would BASIS do anything different with its products?

The Challenges

In prior versions of BBj®, we used an API called "Remote Messages" that passed information between the server and client for character user interface (CUI) displays. Originally, these remote messages passed character data from the server to the client in the form of Java strings stored as bytes. As BBj began to grow in functionality and expand into other world markets, we discovered that an array of bytes was not the same as a string that contained character values with those bytes. This problem first surfaced when our European customers used the Euro character (€). The Euro character displayed differently depending on the character encoding in use, e.g., ISO-8859-15 vs. UTF-8, thus changing what the user saw depending on their configuration. This means that the client and the server must use the same character encoding in order for characters to display correctly on the client and to ensure that characters typed on the client transfer faithfully to the server.

Each Remote Message also required an explicit class to store the data. In some cases a Remote Message required an additional response class, doubling the amount of code in BBj. Correctly creating a new Remote Message meant jumping through many hoops of implementing specific interfaces and methods. This excessive "boilerplate" code required to even implement a single message became difficult to maintain and use. It was too clunky, which lead to further compromises and inconsistencies within the code as the years passed and development progressed.

As we created more messages to handle increasing functionality, the data sent between the client and server grew. This led to communication problems as our customers began to switch to higher latency mobile networks and distributed systems. In a market where the need to stay competitive is paramount, this excessive communication over a high latency network could have detrimental effects on our future products becoming too slow and impractical for distributed mainstream business or personal use.

The compromises inherent in the original Remote Message API led to a poor abstraction in the CUI subsystem between channels and devices. Developers expect an OPEN on two distinct aliases to produce two distinct SysWindow devices. Applications can also open the same device on different channels. Each channel has its own isolated state and operations. To address this issue in the original implementation of CUI, the client maintained a map of CUI instances that resulted in unnecessary sharing between devices on the client. It was not feasible to put this information into the server because of the limitations of the Remote Message infrastructure.

By Aaron Wantuck
Software Engineer

By Adam Hawthorne
Software Engineer

Furthermore, since changing the contents of a Remote Message became such a daunting task, it became necessary for the server to provide the client with an up-to-date view of its server-side environment. Since the source of the state was decoupled from the consumer of that state, the server simply sent all the necessary data instead of only what had changed or what was absolutely necessary. This added to the network overhead as any change to the environment required the server to send another large message to the client so its view of the server-side state would remain consistent.

These issues cried out for a more efficient solution.

The Solution

BASIS experienced similar problems in SYSGUI, the graphical user interface, several years ago. At the time, the engineering team designed a generic library modeled after a Java technology called RMI (Remote Method Invocation). RMI enables developers to write object-oriented code so that objects on different computers can interact in a distributed network. However, the Java RMI design requires a response for every remote method call, which significantly increases the latency of every operation, especially on a slow network. BASIS adapted the architecture of Java RMI to overcome these inherent performance penalties by relaxing restrictions so remote methods may execute without requiring a response. RMI then sends these asynchronous messages in batches, further reducing the network overhead. The library automates as many of these optimizations as possible to reduce the burden on developers.

This simple design of automatically mapping a single method call to a message eliminates the need for extra code and extra objects to implement a single message. Adding a new message is as simple as adding another method to a pre-existing RMI interface and then providing the implementation in the remote class. A developer needs only to consider what operations are actually necessary, and can spend more time perfecting the implementation than on producing the infrastructure.

In CUI, since the individual channels are instances on the server and two channels can easily communicate with the same client device, the client does not have to maintain a "map of instances" any more. If the client requires any information, the server can preemptively send the data to the client just by adding another parameter to a remote method. This completely eliminates the need for the server to continually refresh the client's view of its state. If a particular message requires a particular piece of server state, the server simply sends that single piece of information as a new method parameter. It reestablishes a proper relationship where the server now handles all the device and channel information.

The Results

As a result of all these changes, the user can enjoy faster response times since there is a drastic reduction of network traffic, and BASIS can enjoy cleaner and easier code maintenance. In fact, everyone can enjoy the benefits of this win/win situation.

Neither the server nor the client transfer character data in the native platform encoding. All character data is sent as standard Java String objects, using the ubiquitous UTF-8 character encoding. It is no longer necessary to configure the server and client to have the same character encoding, and both the server and client require less work to translate String objects back and forth into the native platform encoding.

The graph in **Figure 1** shows the results of the new implementation of RMI (red) and the old implementation of Remote Messages (blue). The tests ran a total of fifty times; five sets of five each for RMI and Remote Messages. Each set contained an average time for its set, providing five points for both RMI and Remote Messages. The resultant data was then plotted on the graph, clearly showing marked improvement in the new code. Note that the RMI time never exceeds that of the old remote message time.

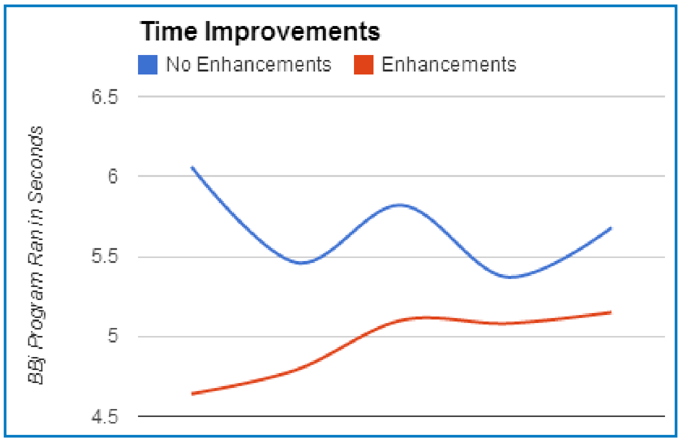


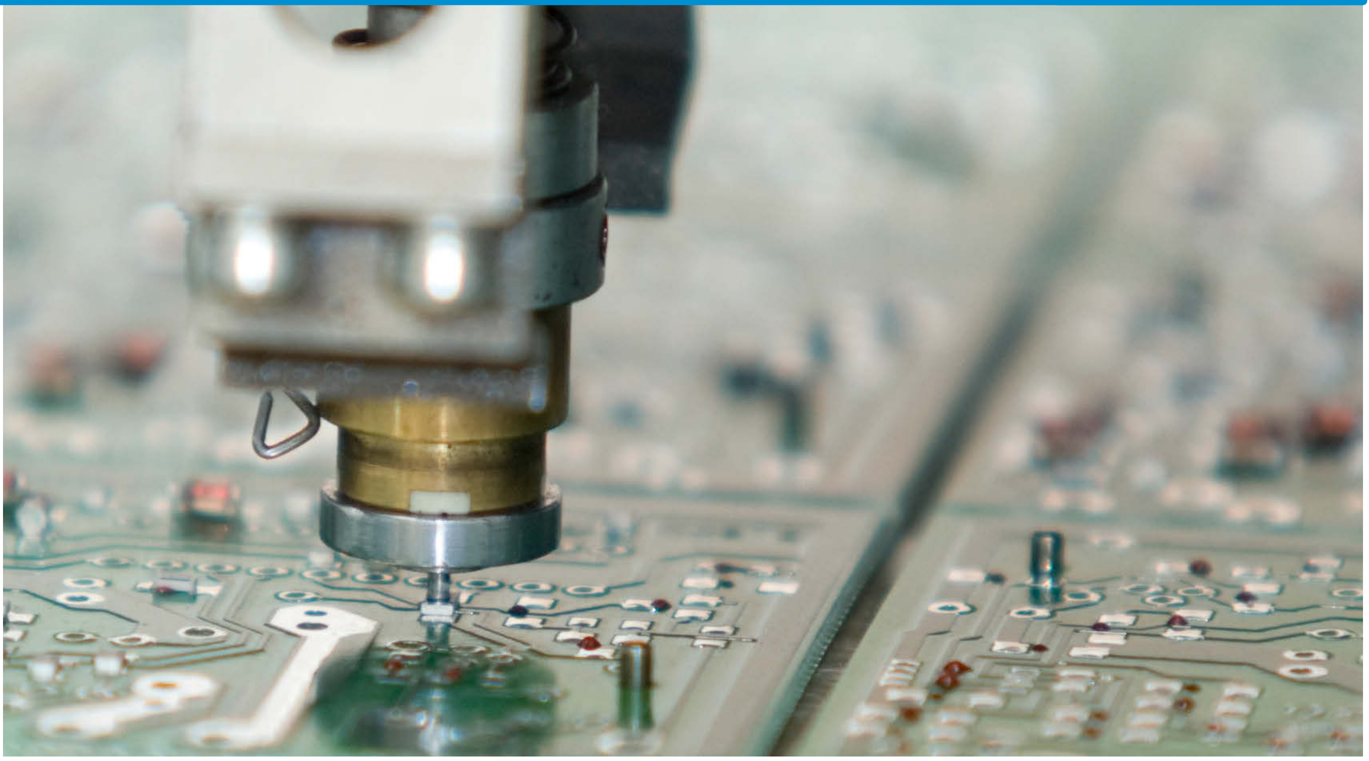
Figure 1. The new implementation of RMI (red) and the old implementation of Remote Messages (blue) in a high latency environment (USA-Europe)

Users can clearly see these enhancements. If they use the SysConsole with a Web Start program or the TermConsole to start any character based application in BBj, they experience an average improvement of around 25% and a much more consistent interface.

Summary

Enhancing product performance is an ongoing activity at BASIS, along with analyzing new ideas for further improvements. Using these future concepts, we can further decrease lag, add more features, and increase the responsiveness of our products, allowing more time for other tasks to run unnoticed. If you think of code as a living entity that is constantly adapting to new stimulus, there is so much more to learn and improve over time. We've seen tremendous adaptations in the medical field, greatly improving from the dark ages to today with such modern discoveries as previously unheard of organ transplants and artificial hearts, to name a few. With software still in its infancy, at just a few decades old, just think what we can do to improve it in another 10 or 20 years! ■

Learn more from these reference materials:
- *The Java EE 5 Tutorial* at links.basis.com/javaee5tutorial
- *Java RMI Tutorial* at links.basis.com/javarmitutorial



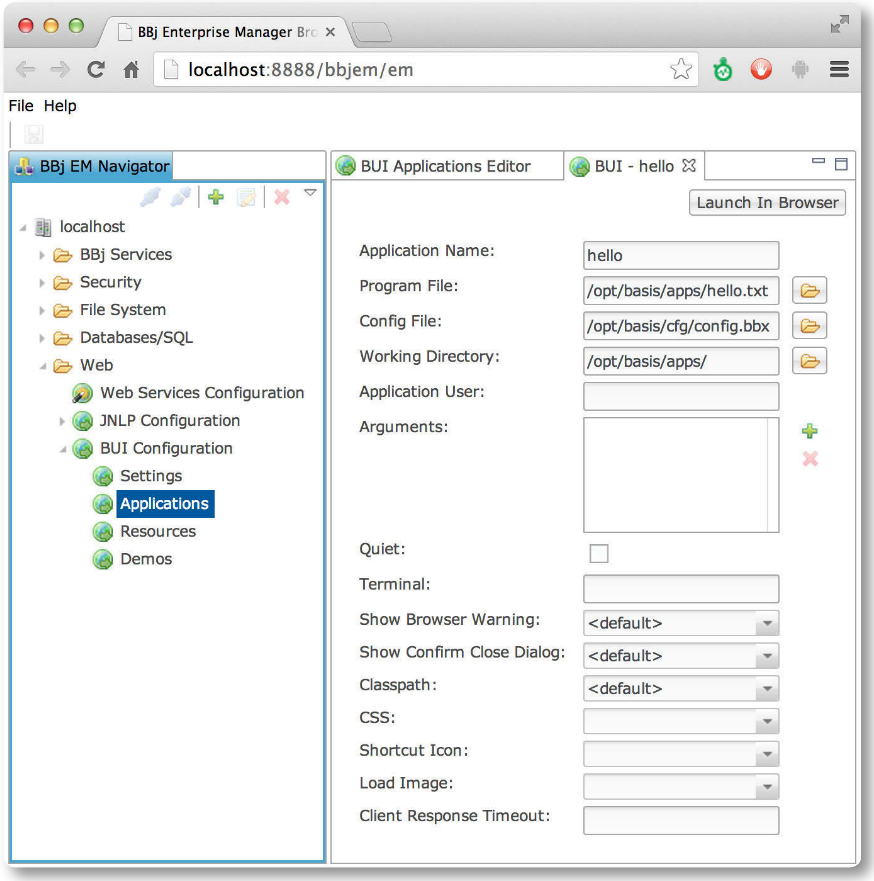
Automate BUI Deployment With the API

Getting your BBj® app running as a BUI web app is a simple matter of spending a few minutes in Enterprise Manager to register the application, define runtime parameters, and (optionally) define a custom CSS file and browser or home screen icon. As of BBj 14.0, BUI programs come with even more optional configuration properties, including load images (a super-fast replacement for your traditional splash screen), and custom end actions that define what the browser does when the BUI app releases. As more configuration parameters become available, developers are expressing the desire to automate this process by programmatically registering and configuring BUI apps. This article looks at how the updated BUI API handles this with ease, allowing developers to register, modify, and manage every aspect of their web-enabled BUI apps.

By Jim Douglas
Software Developer

Getting Started With BUI

The easiest way to get started with BUI is to manually configure a BUI App in Enterprise Manager. See the example in **Figure 1**.



Application Name:	hello
Program File:	/opt/basis/apps/hello.txt
Config File:	/opt/basis/cfg/config.bbx
Working Directory:	/opt/basis/apps/
Application User:	
Arguments:	
Quiet:	☐
Terminal:	
Show Browser Warning:	<default>
Show Confirm Close Dialog:	<default>
Classpath:	<default>
CSS:	
Shortcut Icon:	
Load Image:	
Client Response Timeout:	

Figure 1. A sample BUI App definition in Enterprise Manager

The Enterprise Manager interface is perfect for getting started, and for casually defining a few BUI apps from time to time. However, BUI also has an extensive API for publishing and managing apps. Using the API, you can build automated systems to publish selected apps to the BUI Web Server and update them as needed with new code, CSS, images, and configuration changes. This isn't new; the *BBj BUI: Getting Started* (links.basis.com/buiguide) document has always included a small "Hello World" app (**Figure 2**), along with a sample program that publishes it to the BUI Web Server using the BUI API (**Figure 3**). With the exception of a few small samples like this one, the API documentation was sparse. However, extensive documentation, including full samples, is now available at links.basis.com/bbjappserver and, new for BBj 14.0, links.basis.com/bbjbuimanager.

BUI Development

The BUI API becomes very handy during the development cycle. When you associate image and CSS resources with a BUI app, the app server takes a snapshot of those resources from the moment when they are added. It is very likely that you'll need to adjust any custom CSS associated with a BUI app several times before you are happy with it. When you change your CSS file, you need to inform the app server about the change. That can be done through Enterprise Manager, but it can become tedious when you are iteratively testing CSS tweaks. The easiest approach is to write a small update program that you run every time you edit your CSS file or change a registered image, refreshing the app server before retesting your app.

```
rem ' hello.txt
sysgui = unt
open (sysgui)"X0"
bbjapi! = bbjapi()
sysgui! = bbjapi!.getSysGui()
window! = sysgui!.addWindow(100,100,225,75,"Hello",$00090003$,$$)
window!.setCallback(bbjapi!.ON_CLOSE,"EOJ")
hello! = window!.addButton(1,25,25,75,25,"Hello!")
hello!.focus()
hello!.setCallback(bbjapi!.ON_BUTTON_PUSH,"msgbox")
goodbye! = window!.addButton(2,125,25,75,25,"Goodbye!")
goodbye!.setCallback(bbjapi!.ON_BUTTON_PUSH,"eoj")
process_events
eoj:
release
msgbox:
  i = msgbox(info(1,4),64,fnmode$(info(3,6)))
  hello!.focus()
return
def fnmode$(mode$)
  if mode$="0" then return "Fat Client"
  if mode$="1" then return "Thin Client"
  if mode$="2" then return "Java Applet"
  if mode$="3" then return "Java Web Start"
  if mode$="4" then return "JavaBBjBridge"
  if mode$="5" then return "BUI (Browser)"
  return mode$
fnend
```

Figure 2. A small "Hello World" BUI app

```
rem ' publish.txt
rem ' This assumes that the sample is on the desktop; adjust as necessary
path$ = ""
path$ = env("HOME",err=*next) + "/Desktop/"
if path$="" then path$ = env("HOMEDRIVE") + env("HOMEPATH") + "/Desktop/"
app$ = "hello"
source$ = "hello.txt"
bbjHome! = System.getProperty("basis.BBjHome")
config$ = bbjhome! + "/cfg/config.bbx"
appServer! = bbjapi().getAdmin("admin","admin123").getWebAppServer()
appConfig! = appServer!.makeEmptyAppConfig()
appConfig!.setProgramName(path$ + source$)
appConfig!.setConfigFile(config$)
appConfig!.setWorkingDirectory(path$)
appConfig!.setInterpreterUser(System.getProperty("user.name"))
app! = appConfig!.buildApplication(app$)
appServer!.unpublish(app$,err=*next)
appServer!.publish(app!)
```

Figure 3. A short BUI API program that publishes the sample "Hello World" app

Figure 4 shows an example of this type of program.

```
rem ' update.txt
app$ = "sample"
dir$ = dsk("") + dir("")
css$ = dir$ + app$ + ".css"
img$ = dir$ + app$ + ".png"

appServer! = bbjapi().getAdmin("admin", "admin123").getWebAppServer()
app! = appServer!.getApplication(app$)
appConfig! = app!.toAppConfig()

url! = fnStaticResource!(css$,"text/css")
appConfig!.setStyleSheet(url!)

url! = fnStaticResource!(img$,"image/png")
appConfig!.setLoadImage(url!)

app! = appConfig!.buildApplication(app$)
appServer!.unpublish(app$,err=*next)
appServer!.publish(app!)
print "Updated app ",app$
stop

def fnStaticResource!(filename$,mimetype$)
  iterator! = appServer!.getStaticResources().iterator()
  while iterator!.hasNext()
    resource! = iterator!.next()
    if resource!.getSourceFileName()<>filename$ then continue
    if resource!.getMimeType()<>mimetype$ then continue
    resource!.setSourceFileName(filename$)
    return resource!
  wend
  return appServer!.addStaticResource(filename$,mimetype$)
fnend
```

Figure 4. An update program that refreshes a BUI app definition via the API

New Features

In addition to adding full documentation for the existing BUI API, BBj 14.0 also adds several new features, most of which are accessible through both the interactive Enterprise Manager interface and the programmatic API.

Load Image

The default load image still displays this animated blue and white progress bar,

but you can now define a customized default load image, and custom load images for individual apps. The load image displays almost instantly as the page loads, instead of waiting until after the interpreter associated with the BUI app is up and running. Because it displays so quickly, it is a perfect way to present the user with an application's splash screen or any other customized 'Loading' indicator.

End Action

We've had many requests for developer-defined application termination actions in place of the default of just replacing the application in the browser with a "Click to reload application" message. As of BBj 14.0, custom end actions are definable, both at the global default level and for individual apps.

There are several kinds of end actions. You can continue to clear the browser window (the original and still default end action), but show a custom message in place of the default message. Or you can chain to another BUI app, perhaps one that acts as a main menu, or chain to a particular URL. For more flexibility, BBjBuiManager even allows for changing the end action dynamically in a running BUI app. In all cases, the end action takes effect when the app terminates, typically by executing a RELEASE statement.

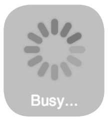
Busy Indicator

Traditional desktop applications typically indicate that they are busy by setting the wait cursor. While this still works with BUI apps in desktop browsers, mobile browsers don't have cursors so they need some other way to let the user know that they are busy. The most common way to indicate that a browser-based application is temporarily busy is to dim the screen and display an animation with a short message to let the user know what is happening. Developers can now easily accomplish this with the new Busy Indicator, as the code in **Figure 6** demonstrates.

```
busy! = bbjapi().getBuiManager().getBusyIndicator()
busy!.setText("Busy...")
busy!.setVisible(1)
rem -- do a bunch of work --
busy!.setVisible(0)
```

Figure 5. Code to create, customize, and display a BUI busy indicator

When the code sets the busy! object **`visible`**, the user sees the animated busy indicator shown here to the right.



Summary

In the past, publishing and managing your BUI apps required a great deal of interaction with Enterprise Manager's BUI Configuration screens. To streamline and help automate this process, BBj now offers a robust API that can do everything Enterprise Manager does – from first-time registration of a BUI app to modifying every available configuration parameter. In addition, developers can implement new features such as the customizable load image, dynamic end action, and the busy indicator interactively or programmatically. The API can save you a lot of time and gives your mousing hand a much-needed rest. Give it a try today! ■

links.basis.com/**13code**

- Read *BBj BUI: Getting Started* at links.basis.com/buiguide
- For more on the BBjBusyIndicator and custom end actions, see *BUI to the Max – Amp Up and Fine-Tune* at links.basis.com/13bui

BBj DBMS Now Leaves PRO/5 Data Server in its Dust!

Nearly every application used in the business world provides multiple users with simultaneous access to the application and data. As business needs grow, it is vital that the application reliably scales with regard to user interface and data access performance. BASIS recently took a great step forward, significantly improving the performance of the BASIS DBMS, and thus, SQL data access when simultaneously accessing data files from multiple users or other programs.

The Problem

Traditionally, both the PRO/5® filesystem and the BBj® BASIS DBMS serialized reads on the same file, meaning that only one process could read from the file at a time. Since reads are extremely fast, this is not normally a performance issue. However, it can become a bottleneck if several processes perform a large number of reads on a single file simultaneously. The PRO/5 filesystem has a 'multi-reader' setopts bit to enable shared read locking on files, providing some scaling for concurrent reads. Unfortunately, the use of operating system file locks means that shared read locking is not balanced with writes and can cause writes to "starve" and not complete in a timely manner. In this case, the setopts bit cannot be used safely by most customers.

The Solution

Beginning with the BBj 13.10 preview of BBj 14.0, the BASIS DBMS scales concurrent reads on the same file so multiple readers can access the same file at the same time, limited only by the operating system and hardware. Rather than using the shared read locking at the operating system level (as with the PRO/5 filesystem), the BASIS DBMS uses internal fair read-write locking. This allows BBj to safely and transparently scale reads without starving writes. As a result, the BASIS DBMS is the ideal choice for customers who need the PRO/5 filesystem shared read or multi-reader setopts bit set because not only does it eliminate the potential for write starvation, it is always enabled and performs significantly better than the PRO/5 filesystem with the shared read setopts bit enabled, providing a fully transparent performance improvement for all customers.

The Benchmarks

Demonstrating and testing read scaling was very straightforward. We ran multiple copies of a simple program simultaneously that iterated through a large file. While this may not have necessarily reflected a real-world use case for customers, it clearly showed the benefits of read scaling during multi-user access. **Figure 1** shows the results of this simple test.

By Chris Hardekopf
Software Engineer

By Jeff Ash
Software Engineer

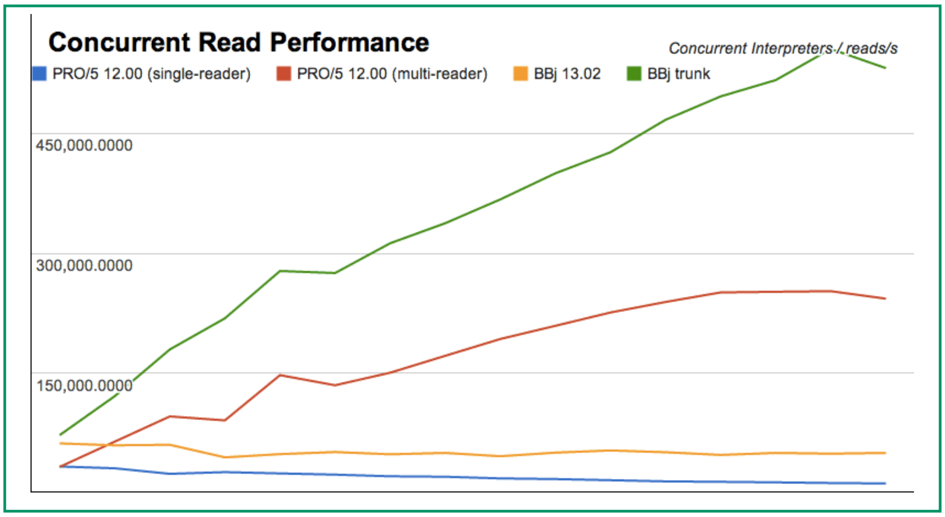
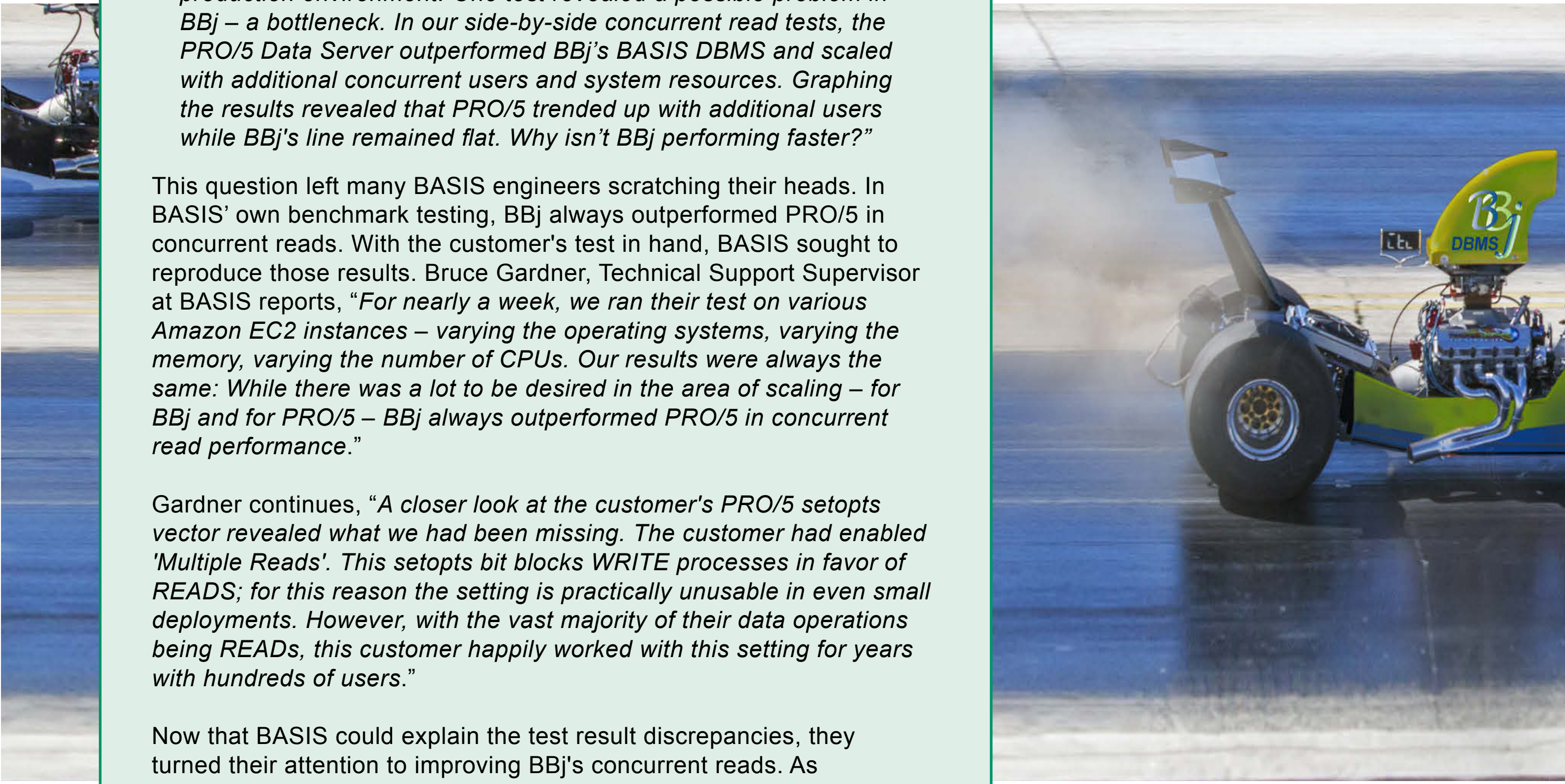


Figure 1. Concurrent read testing results, with the best performance from the new code shown in green

To get a good feel for the impact of the new read scaling capability, we ran our tests against both PRO/5 and BBj. Note that traditional PRO/5 and BBj concurrent read performances are rather flat as represented by the blue and orange lines. In other words, as we ran more copies of the test program, each reader performed slower in a linear fashion. For example, with two concurrent readers, each individual reader was approximately half as fast, with three it took three times as long, etc. With PRO/5 shared read locking (red line) and the new BASIS DBMS implementation (green line), the reads scale to the number of readers, up to operating system and hardware limitations. Not only does BBj eliminate the possibility of write starvation, it is also significantly faster than PRO/5 as a result of the improvements made to the filesystem. You can see this performance on your system today with BBj 13.10 which first previews this BBj 14.0 feature.

Summary

As business needs grow, it is vital that core enterprise applications and their required data access scale as well. BBj 14.0 takes another huge step forward providing this scaling in the area of concurrent read access to the data, all without any configuration or application changes. ■ links.basis.com/**13code**



A Case Study

As with many BBj features and improvements, BASIS' work on concurrent read performance began with a question posed by a customer.

“Preparing to migrate a PRO/5 deployment to BBj, our company's engineering staff performed a series of tests in a simulated production environment. One test revealed a possible problem in BBj – a bottleneck. In our side-by-side concurrent read tests, the PRO/5 Data Server outperformed BBj's BASIS DBMS and scaled with additional concurrent users and system resources. Graphing the results revealed that PRO/5 trended up with additional users while BBj's line remained flat. Why isn't BBj performing faster?”

This question left many BASIS engineers scratching their heads. In BASIS' own benchmark testing, BBj always outperformed PRO/5 in concurrent reads. With the customer's test in hand, BASIS sought to reproduce those results. Bruce Gardner, Technical Support Supervisor at BASIS reports, “*For nearly a week, we ran their test on various Amazon EC2 instances – varying the operating systems, varying the memory, varying the number of CPUs. Our results were always the same: While there was a lot to be desired in the area of scaling – for BBj and for PRO/5 – BBj always outperformed PRO/5 in concurrent read performance.*”

Gardner continues, “*A closer look at the customer's PRO/5 setopts vector revealed what we had been missing. The customer had enabled 'Multiple Reads'. This setopts bit blocks WRITE processes in favor of READs; for this reason the setting is practically unusable in even small deployments. However, with the vast majority of their data operations being READs, this customer happily worked with this setting for years with hundreds of users.*”

Now that BASIS could explain the test result discrepancies, they turned their attention to improving BBj's concurrent reads. As discussed above, the research was very fruitful. Not only was the BBj concurrent read bottleneck removed, but now BBj's BASIS DBMS far outperforms and scales better even than the PRO/5 Data Server with the multiple read setopt bit enabled.

Bruce Gardner
Technical Support Supervisor

So how much do you have left after buying a little app?
Ten bucks and 9.9999999999999996 cents!
$11
- 90¢ app
???

Mix 'n Match Data Structures Between BBj and Java

One of the key advantages of BBj® is its seamless integration with the Java platform on which it runs. Developers can use Java objects within their BBx® applications as though they were native BBj objects. But on the rare occasion, a developer must use some of the more advanced features of BBj to deal with a semantic mismatch one may encounter between BBx, a language with a 30-year lineage, and Java, having a far briefer teenage lineage. This article discusses some of the tips and tricks as well as some recent improvements to the BBj version of the BBx language and in the BBj API that provide help in these situations.

Character Encoding

When two countries with different languages share a border, there is always an area where the two cultures clash. The issues surrounding character encoding between BBx and Java is similar at multiple levels.

By Adam Hawthorne
Software Engineer

The term "character encoding" would be better called "character transcoding." A character set (or "charset") defines a mapping between a range of integers and a set of symbols that each represents an atomic unit of written language, known as a "character." A character encoding defines the algorithm by which the integers from a given charset are then encoded as a sequence of 8-bit bytes, and also by which a sequence of bytes are then decoded back into valid integers in that charset.

The engineers who created Java chose to use the Unicode character set as their mapping between integers and symbols. Unicode strives to assign a number to every distinct character. Internally, Java must represent the integer associated with each character as a sequence of bytes. Java does this by representing each integer as one or more 16-bit characters in a character encoding known as UTF-16. These characters are held in an object called a **`java.lang.String`**, or a "Java String."

BASIS created BBx before Unicode was even a twinkle in its creators' eyes, in a simpler time when 64KB really was enough for anyone. The idea of using two bytes for every character would have been an unthinkable waste of space, and the state of the art was to use an encoding in which, at most, 256 character symbols could each be represented by a single byte. This relationship between bytes and characters became an assumption on which much of the language and tools depends even today. The familiar string variable X$, literal string constants such as "**`abcdefg`**," and hexadecimal string constants such as **`$61626364656667$`** are all stored internally as a sequence of 1-byte values.

The boundary between Java and BBx where a BBx string of bytes is converted into a Java String is hidden for the most part by using the character encoding defined by the operating system. On UNIX systems, Java uses the contents of the **LC_ALL, LC_CTYPE** or LANG environment variables to determine the default character encoding. On Windows systems, Java uses the user's Control Panel preferences.

The Java notion of a capital "C" **Charset** includes an encoding and decoding algorithm to convert from a sequence of bytes that represent the integers corresponding to the characters from that charset into a sequence of 16-bit integers that represent those same characters in Unicode (and back).

When passing a byte-per-character BBx string value into a Java method that takes a Java String, BBj uses the default **Charset** to convert from those bytes into a Java String. When returning a Java String from a Java method, BBj uses the default **Charset** to convert from the String back into bytes.

This implicit algorithm has the potential to introduce unintended behavior. However, combined with the Java API, BBjAPI contains several methods to defeat or modify the implicit behavior by which we can smooth over these differences between the Java language and the BBx language. See a comparison of the different methods in **Figure 1**.

METHOD	DESCRIPTION	USE/RATIONALE
BBjAPI::asBytes**(string)***	Obtain an instance of the Java byte[]object that represents a BBx string.	There may be two methods with the same name; one that takes a java.lang.String and one that takes a byte[]. Use the result of this method as an argument to select the one that takes a byte[], otherwise BBj will use the method that takes a java.lang.String. BBj will automatically convert any BBx string value passed to a method that takes a java.lang.Object to a java.lang.String. This conversion can result in a loss of data, especially if the data is not textual in nature. Use the result of this method to avoid any automatic conversion for a java.lang.Object parameter type.
BBjAPI::toLocal**(string)**	Obtain the unmodified encoding for characters obtained from a java.lang.String.	BBj converts bytes that do not map to a valid character in a particular charset to a special range of characters in Unicode called the “Private Use Area”. These characters appear as UTF-16 values 0xE0FF-0xE1FF. Using this method prevents a String containing characters in this range from being translated back into potentially valid byte values. This might be important if the data in the String represents something other than character data and should not be modified in any way. Each encoding has a unique byte sequence to represent a character that does not have a valid encoding. For Cp1252, this is the byte value 0x3F, which corresponds to the question mark character‘?’.
BBjAPI::toUnicode**(string)**	Obtain a pure Unicode **java.lang.String for a byte[]** or BBx string.	BBj converts bytes that do not map to a valid character in a particular charset to a special range of characters in Unicode called the “Private Use Area”. These characters appear as UTF-16 values 0xE0FF-0xE1FF. Using this method prevents this translation from occurring, and any invalid bytes will be replaced with the replacement character value 0xFFFD. This could be useful for determining that a certain sequence of bytes is invalid character data.
new String(byte[], String)	Create a java.lang.String interpreting the byte[] argument with a specified encoding named by the String parameter.	Use this if you are receiving byte[] data from another source and the byte[] might have been generated using a different charset than the platform charset. E.g., consider a situation where a Windows computer generates byte data using the Cp1252 charset, and inserts that data into a data file. Then, the data must be used on a Linux server using the ISO-8859-15 charset. Use this method to transform that data into a java.lang.String: str! = new String(x$, “Cp1252”). Then, use toLocal(str!) to transform the java.lang.String back into a BBx string.
String.getBytes(String)	Create a byte[] or BBx string value using a particular encoding named by the String parameter	This method creates a sequence of bytes using a specified character encoding. For instance, assume one must create a document for a partner using the UTF-8 character encoding. Once the contents document exist in a string x$, one might use the following code to obtain the UTF-8 bytes: y$ = toUnicode(x$).getBytes(“UTF-8”). Then, write the string y$ to a file.

Figure 1. Summary of the different methods and their uses

Numbers and “Business Math”

One of the original motivations for creating Business BASIC was to address the sometimes-counterintuitive behavior of traditional binary floating point numbers. Binary floating-point data types favor speed and a compact representation using base 2 arithmetic. Numbers using so-called "business math," also known as "binary coded decimal" (BCD), favor predictable, intuitive results using base 10 arithmetic that match calculations made by hand. Since BBx uses BCD operations for real number calculations, traditional BBx programs avoid the confusion that arises from the behavior of binary floating-point calculations.

The Java programming language, on the other hand, derives much of its syntax and semantics from the "C" family of languages, which exposes the native binary floating-point operations of the underlying hardware via its primitive `float` and `double` data types. Now that BBj provides access to the massive Java ecosystem, the differing behaviors of the traditional and business approach to real numbers may surprise BBx developers when using libraries that expect Java `double` or `float` types.

BBj once again smooths over many these differences by automatically converting from its internal BCD representation to the binary floating point representation. However, there are two inconsistencies BBx developers should be aware of:

1. The Java `float` and `double` types cannot store exact values for many common decimal values. When converting from one of these floating point types into a BBx numeric type, you may wish to pass it to the ROUND() function first. Consider the output from this program:

```
PRECISION 16
a = new Double("10.1")
PRINT a
a = ROUND(new Double("10.1"), 2)
PRINT a
```

Output:
```
10.09999999999999996
10.1
```

The value 10.1 cannot be represented as a finite sequence of numbers in base 2. Neglecting to take that into consideration can produce confusing results.

2. Java does not allow automatic conversions (called "narrowing conversions") from a `double` to a `float`. BBj strives for consistency with Java by disallowing an automatic conversion from a BBx numeric type directly to a `float` However, the CAST() function allows a program to perform this conversion:

```
DECLARE float a!
a! = CAST(float, 10.1)
print a!
```

Output:
```
10.1000003814697266
```

For in-depth information describing the gory details of binary floating point representation, visit tinyurl.com/czqwkr9.

Java Arrays

With more data comes more ways to interpret that data, even with the most simple of compound data types: the array. BBx implements an array type as a contiguous block of memory with metadata that provides the information about the dimensional structure of the array. Java arrays are implemented as nested "arrays of arrays," where each element in any intermediate arrays may refer to an individual array. See both storage layouts illustrated in **Figure 2**.

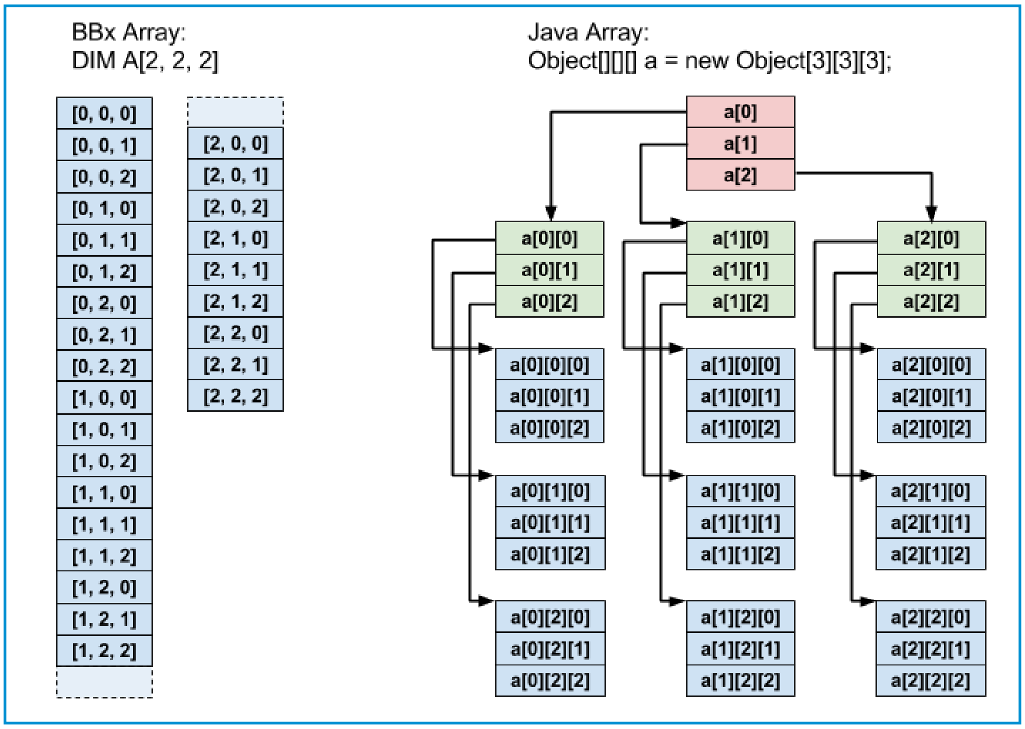


Figure 2. BBx and Java array examples

Because of this mismatch, BBj converts from Java arrays to BBx arrays to help when interacting with Java. A method or object variable that returns a Java array may be assigned to a BBx object array variable, e.g.:

```
locales![] = java.util.Locale.getAvailableLocales()
```

The Java array is treated as a 1-dimensional array, regardless of the number of dimensions used to create it originally. The elements of the BBx array may contain other Java array values.

BBjVector

BBj also provides the **BBjVector** class to bridge the gap between Java arrays and BBx syntax. Use BBJAPI().makeVector() to obtain an instance of a BBjVector. BBjVector can be passed to any method that receives a Java array parameter, as long as the individual elements in the **BBjVector** are all assignable to the component type of the array. For example, one may use the **String.format(String, Object[])** method to create a formatted string from Java. The following program:

```
vec! = bbjapi().makeVector()
vec!.addItem(new Double("68.2"))
vec!.addItem("Fahrenheit")
print String.format("The temperature is %.1f degrees %s.", vec!)
```

produces the result:

```
The temperature is 68.2 degrees Fahrenheit.
```

The parameter **vec!** is automatically converted into an **Object[]**.

Java Class and Package Names

Java packages, classes, methods, and fields may all begin with the underscore '_'. It is now possible to refer to these Java productions from a BBj program. Here is the motivating example, supported in BBj 13.0 and above:

```
USE com.microsoft.schemas.exchange.services._2006.messages.ExchangeService
```

This feature also enables the use of an underscore at the beginning of a standard variable name:

```
_s$ = "Hello, world!"
_n = 10
_o! = new Object()
```

Summary

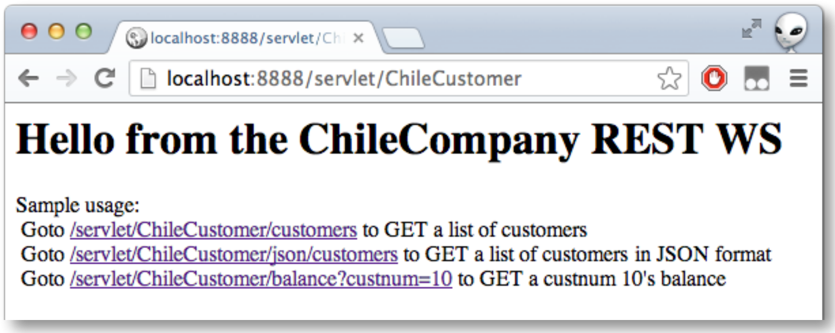
BBj does most of the heavy lifting to hide the inconsistencies between BBx and Java. Most BBx applications never need to concern themselves with the level of detail described here. But if these differences ever do cause your application or your data to misbehave, this view behind the curtain can give you the tools you need to put it back in line. ■ links.basis.com/**13code**

Sit back and enjoy a 30-minute presentation with BASIS!

links.basis.com/javabreak

BBjServlets Serving Web Content

If you attended TechCon2013, you saw a demonstration that provided a REST-like Web Service for access to our Chile Company database. Servlets in general help to overcome latency and bandwidth limitations through a platform independent protocol implementation because all the work happens at the server and only the answer traverses the network to the client. This particular BBjServlet demo showed a BBj® program implemented as a BBjServlet, leveraging the HTTP protocol to provide responses from individual URLs to various clients. The demo's functionality included the ability to get a list of customers in multiple formats, including the ever-popular JSON (www.json.org) format, and the ability retrieve a particular customer's balance as illustrated in **Figure 1**. This article delves deeper into the demo's source code, providing a working example of one possible way to take advantage of the BBjServlet's capabilities by providing web-based content to a variety of disparate clients.



localhost:8888/servlet/ChileCustomer

Hello from the ChileCompany REST WS

Sample usage:
Goto /servlet/ChileCustomer/customers to GET a list of customers
Goto /servlet/ChileCustomer/json/customers to GET a list of customers in JSON format
Goto /servlet/ChileCustomer/balance?custnum=10 to GET a custnum 10's balance

Figure 1. A web browser accessing the published service with sample URLs

Scrutinizing the Source

Let's take a closer look at the code necessary to run this demo.

The servlet implementation, an excerpt of which appears in **Figure 2**, begins by creating a BBj custom object, then obtains a servlet data object and registers for the ON_WEB_CONNECTION event in order to handle the incoming HTTP requests.

By Brian Hipple
Quality Assurance Supervisor

```
declare BBjServletData sData!

myServlet! = new myServlet()
sData! = BBjAPI().getServletData()
sData!.setCallback(sData!.ON_WEB_CONNECTION, myServlet!, "customer")
```

Figure 2. The beginning of the servlet implementation code

As covered in a previous Advantage article, *Rest Easy - End Your WSDL Struggles* (links.basis.com/12rest), clients access the servlet with an HTTP GET request. Therefore, the customer method of our servlet is able to ascertain what information the client is requesting by examining the HTTP request's path information. The BBjServlet also supports URL parameters, providing an easy way for a client to refine further their communication with the servlet. The third sample URL from **Figure 1** shows how the client may include a parameter to specify a unique customer in the database by providing the desired customer number.

The custom **`myServlet!`** class, referenced in the Callback code in **Figure 2**, is accessed whenever a client makes a connection to our published application. The beginning of the **`myServlet!`** class definition is shown in **Figure 3**, and illustrates how the class's customer method obtains the HTTP request and creates an HTTP response object based on the provided BBjServletEvent.

```
REM ================================================================
REM Servlet Class
REM ================================================================
class public myServlet

    method public void customer(BBjServletEvent p_event!)

        declare BBjHttpResponse response!
        declare BBjHttpRequest request!

        LET chan = UNT
        request! = p_event!.getHttpRequest()
        response! = p_event!.getHttpResponse()
        response!.setContentType("text/html")
        open(chan)"JSERVLET"
        finished = 0
```

Figure 3. The beginning of the myServlet class

```
REM ==========================================
REM JSON Customers
REM ==========================================
if (request!.getMethod() = "GET" and request!.getPathInfo()="/json/customers" ) then
    REM Get customers
    customersMap! = #getCustomers()
    it! = customersMap!.keySet().iterator()

    json$ = "{ ""customers"" : ["
    while it!.hasNext()
        json$ = json$ + " {""name"":""" + str(it!.next()) + """},"
    wend
    json$ = json$(1,len(json$)-1)
    json$ = json$ + "] }"
    print(chan) json$
    finished = 1
endif
```

Figure 4. The servlet code that sends the list of customers to the client

The servlet handles each client's HTTP request by performing the appropriate SQL query to the Chile Company's customer table in order to retrieve the requested data. The servlet then builds a response and sends it back to the client by writing to the **`JSERVLET`** channel. **Figure 4** shows how the servlet handles a GET request for a list of the Chile Company customers in JSON format.

The code begins by calling a **`getCustomers()`** method, defined later in the source code, that executes an SQL statement and returns a java HashMap data structure with the list of customers. It then builds the string result by iterating over the HashMap, adding the customers into a JSON array object. When it has finished creating the string representation of the JSON response, the servlet sends the result to the client by printing the string to the JSERVLET channel.

Making Your Own Servlet

To make a BBjServlet available to clients, follow these three easy steps:

1. **Create the application configuration.** The BBjApplication object requires configuration information such as the program to run, which config file to use, and the working directory. The BBjAppConfig object encapsulates this program information and stores it so that the program can run as a servlet in an automated fashion. This is much like an autorun program or scheduling task.
2. **Obtain the servlet registry**. The BBjServletRegistry manages all currently running servlets and assists in publishing BBjAppConfig objects at specific paths. The registry is obtained from the BBjAdmin, in which administrative rights are required to add and remove servlets to and from the built-in Jetty Web Server.
3. **Publish the servlet.** Simply select a path, which starts with "/servlet/," and publish the application using the BBjAppConfig and BBjServletRegistry objects obtained in steps #1 and #2. As clients connect to the service, new interpreters start up to handle the request and are pooled, depending on load. As the load declines, these interpreters will shut down. Licenses are only checked out during request handling.

```
declare BBjAdmin admin!
declare BBjServletRegistry registry!
declare BBjConfig config!
declare BBjAppConfig application!

REM create
config! = BBjAPI().getConfig()
application! = config!.makeEmptyAppConfig()
application!.setWorkingDirectory(dsk("")+dir(""))
application!.setConfigFile(System.getProperty("com.basis.server.configDirectory")+"/config.min")
application!.setProgramName("chileCustomerService.src")

REM obtain
admin! = BBjAPI().getAdmin("admin", "admin123")
registry! = admin!.getServletRegistry()

REM publish
registry!.unpublish("/ChileCustomer", err=*next)
registry!.publish("/ChileCustomer", application!)

REM Goto http://localhost:8888/servlet/ChileCustomer to test
release
```

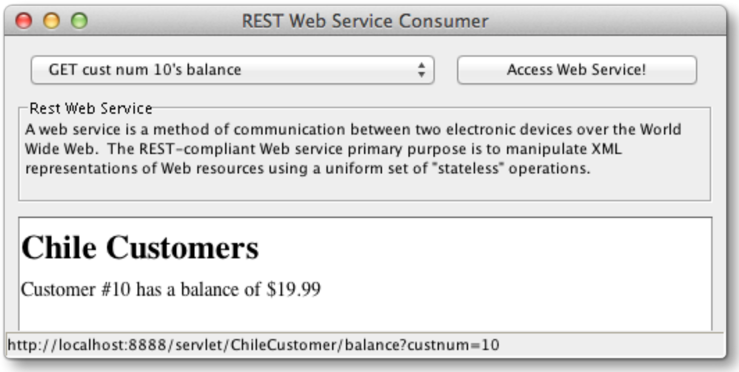
Figure 5. The code required to publish a servlet

Figure 5 shows the source code that encompasses all three steps.

Accessing Your Servlet

Any HTTP enabled application or tool can perform servlet invocation, including web browsers as shown in **Figure 1**. For the scope of this article, we are going to write the client in our favorite computer language, BBj. The BBj application provides a user interface shown in **Figure 6**, which allows the user to select the desired Web Service function to invoke from a listbox.

Upon user selection of a servlet function, the appropriate request creates an HTTP client using the functionality provided in the Java org.apache.commons.httpclient package. The request then executes, and the response returns and is displayed in a static text field, as shown in the code excerpt in **Figure 7**.



A web service is a method of communication between two electronic devices over the World Wide Web. The REST-compliant Web service primary purpose is to manipulate XML representations of Web resources using a uniform set of "stateless" operations.

Chile Customers

Customer #10 has a balance of $19.99

http://localhost:8888/servlet/ChileCustomer/balance?custnum=10

Figure 6. Our Web Service client written in BBj

```
CASE 3
   request!= new org.apache.commons.httpclient.methods.GetMethod(getBalanceURL$)
   client!.executeMethod(request!)
   returnString$=request!.getResponseBodyAsString()
   restView!.setText(returnString$)
   restStatus!.setText(getBalanceURL$)
break
```

Figure 7. An excerpt of the client code's SWITCH statement that communicates with the Web Service

Summary

As you can see, it takes very little code to create a servlet program and associated client programs. The fact that it is stateless and is built on a well-known and easy-to-use HTTP protocol makes it the most attainable way to share data and business rules with disparate systems. When your next project calls for applications to communicate over HTTP, think BBjServlet!

links.basis.com/13code

For more information, refer to *Rest Easy - End Your WSDL Struggles* at links.basis.com/12rest

Prodin Blends BASIS With Chocolate for Sweet Success

The Hilversum-based BASIS partner Prodin Business Solutions, founded in 1975 as InfoStore Nederland B.V., has a history of delivering software development and tailor-made professional services for customers in the Netherlands and beyond, reaching back to the days of MAI. But that legacy is nothing compared to the tradition that their customer Stollwerck has in its DNA.

Stollwerck, founded in 1839, is a household name for chocolate and sweets which every German's tastebuds, adult and child alike, remembers dearly. In 2011, when Baronie/Sweet Products N.V. acquired Stollwerck, Prodin stepped onto the scene. In a move to strengthen and unify the IT landscape of the whole group, Baronie had Prodin replace the SAP system Stollwerck had been using.

Long-term Prodin and Baronie Partnership

Prodin and Baronie have been working together since 1988, when Prodin sold the standard BBx® Prodin-P2 solution on a UNIX platform to Baronie de Heer BV in the Netherlands. During the implementation, Baronie added more specific functionality to cover their needs. In the early nineties, Baronie took over the development themselves and virtually made their own, highly individualized version of Prodin-P2. This became their standard ERP package which is still in use today.

In 2010, Baronie contacted Prodin to request programming support for their tailor-made P2 solution. At that time, the solution served five Baronie sites in Belgium and the Netherlands, including four production sites and a distribution warehouse. When Baronie took over Stollwerck in 2011 from Barry Callebaut, another major player in the European and global sweets market, Baronie examined the possibilities of forming a common effective IT platform in order to bring the Stollwerck processes in line with the ones used at Baronie, and to generally optimize intercompany processes. Specifically, Baronie intended to reduce the complexity of the processes which were in place at Stollwerck.

By Patrick Schnur
European Marketing/PR

Prodin
Business Solutions

Henk van Heemskerk
Director

Founded in 1975 as "InfoStore Nederland B.V.", **Prodin Business Solutions** introduced a suite of business software for MAI which they have extended and improved ever since. Today, the current product generation Prodin-P3 is a suite of ERP modules suitable for a wide range of companies, including production companies (project-based or serial production), wholesalers, as well as service and maintenance suppliers. In 2003, Prodin was awarded the SAP Innovation award for the application of the SAP Netweaver technology on Prodin-P3 for their customer Royal Schelde. Completed with specialized modules such as Customer Relationship Management (CRM) or the mobile solution "Prodin Business Workplace" coined for the on-site use by service and maintenance personnel, Prodin-P3 is a state-of-the-art ERP solution which a wide customer base of SMEs in many European countries put their trust in.

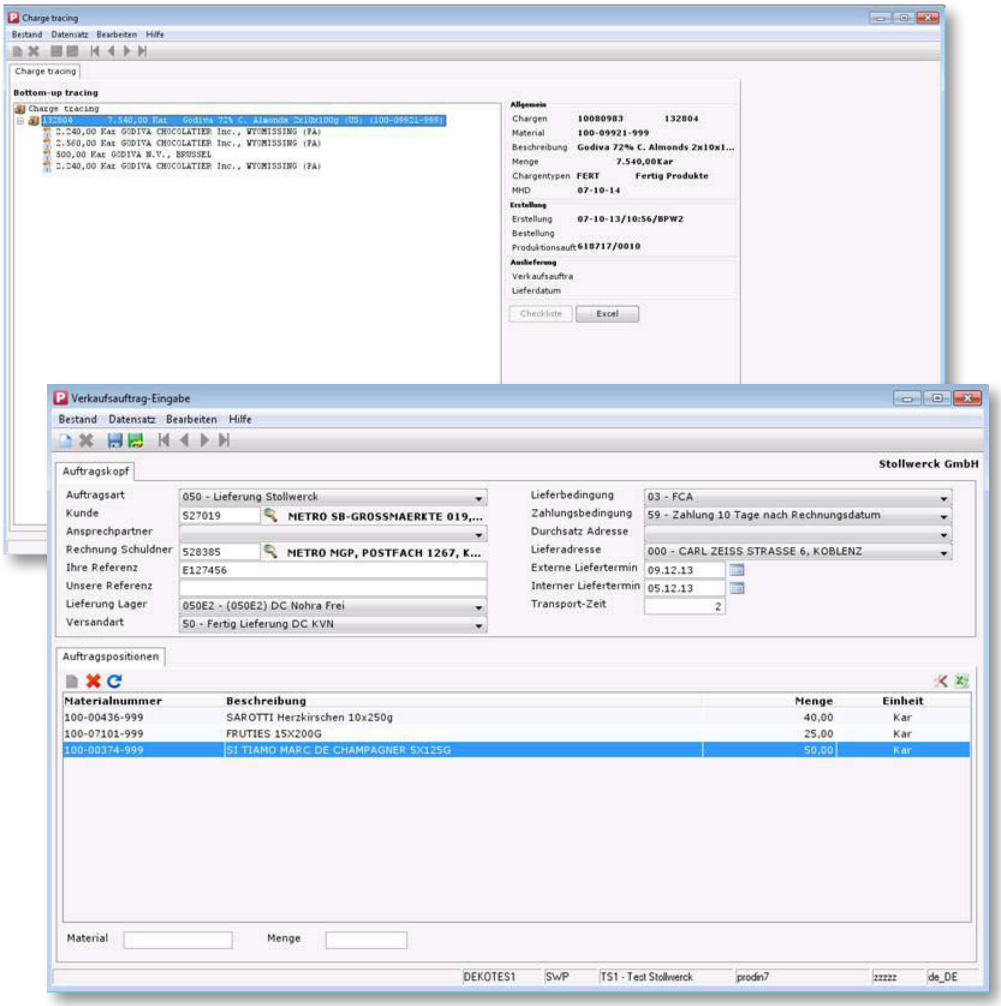
Prodin has been a BASIS partner since the 1980s, embracing all generations of BASIS technologies to supply their customers with the latest and greatest software technology, culminating in 2012 with Prodin-P3 support on iPads and tablet PCs, thanks to BASIS' Browser User Interface (BUI). www.prodin.nl

At the time, Stollwerck had an IT landscape that consisted of a hosted SAP 4.6 environment combined with tailor-made software at all their production sites and warehouses in Germany, France and Belgium. They ran the standard SAP modules for inventory management, production planning, quality management, sales/CRM, business data warehouse and finance/accounting, enhanced with a solution to map to the specific pricing and trade terms of the German market.

Streamlining the IT Landscape: Prodin vs. SAP

In order to streamline the IT systems, Baronie had two options to consider: They could either implement SAP on both sides, or they could make Stollwerck use the Prodin solution, which covered all of Baronie's needs and which they had been using successfully for more than 20 years.

Since Prodin is experienced in both SAP as well as analyzing and re-engineering business processes, they carried out an analysis to determine the gap between processes and systems. The study found that P2 could only have supported the Stollwerck processes with a huge and costly impact on programming. On the other hand, the proverbial complexity of SAP could not be reduced to an acceptable level so that Baronie could have worked with the SAP system either.



Bottom-up tracing

```
Charge tracing
  132804        7.540,00 Kar   Godiva 72% C. Almonds 2x10x100g (US) (100-09921-999)
    2.240,00 Kar GODIVA CHOCOLATIER Inc., WYOMISSING (PA)
    2.560,00 Kar GODIVA CHOCOLATIER Inc., WYOMISSING (PA)
    500,00 Kar GODIVA N.V., BRUSSEL
    2.240,00 Kar GODIVA CHOCOLATIER Inc., WYOMISSING (PA)
```

Allgemein

Chargen	10080983	132804
Material	100-09921-999	
Beschreibung	Godiva 72% C. Almonds 2x10x1...	
Menge	7.540,00Kar	
Chargentypen	FERT	Fertig Produkte
MHD	07-10-14	

Erstellung

Erstellung	07-10-13/10:56/BPW2
Bestellung	
Produktionsauft	618717/0010

Auslieferung

Verkaufsauftra	
Lieferdatum	

Stollwerck GmbH

Auftragsart	050 - Lieferung Stollwerck	Lieferbedingung	03 - FCA
Kunde	527019 METRO SB-GROSSMAERKTE 019,...	Zahlungsbedingung	59 - Zahlung 10 Tage nach Rechnungsdatum
Ansprechpartner		Durchsatz Adresse	
Rechnung Schuldner	528385 METRO MGP, POSTFACH 1267, K...	Lieferadresse	000 - CARL ZEISS STRASSE 6, KOBLENZ
Ihre Referenz	E127456	Externe Liefertermin	09.12.13
Unsere Referenz		Interner Liefertermin	05.12.13
Lieferung Lager	050E2 - (050E2) DC Nohra Frei	Transport-Zeit	2
Versandart	50 - Fertig Lieferung DC KVN		

Materialnummer	Beschreibung	Menge	Einheit
100-00436-999	SAROTTI Herzkirschen 10x250g	40,00	Kar
100-07101-999	FRUTIES 15X200G	25,00	Kar
100-00374-999	SI TIAMO MARC DE CHAMPAGNER 5X125G	50,00	Kar

Material | Menge

DEKOTES1 | SWP | TS1 - Test Stollwerck | prodin7 | zzzzz | de_DE

Figure 1: Lot tracing (top) and sales order entry (bottom) in Stollwerck's Prodin-P3

After a thorough and meticulous decision-making process, Baronie finally decided on a third option. As the centerpiece of the new landscape, Prodin was to implement an out-of-the-box version of P3 (**Figure 1**) at both Stollwerck and Baronie, which is based on Visual PRO/5® and BBj®, and runs Windows Server 2008 on hardware located in Cologne, Germany. A Cognos replication database will be hosted at the Prodin data center in Haarlem, Netherlands.

STOLLWERCK

*Founded in 1839 in Cologne by Franz Stollwerck, the company first produced cough drops very successfully, expanding its range to chocolate and marzipan, and later, gingerbread. The company's history very much reflects the eventful European history. After a phase of expansion in the 19th century with factories opened in London, Vienna, Romania and the Austro-Hungarian empire, Stollwerck suffered from the depression in the late 1920s and the consequences of World War II. But in 1947, business successfully restarted. Between 1948 and 2000, Stollwerck acquired numerous renowned competitors, expanded its market share and established the company as a global brand.

Today, Stollwerck runs five factories in Germany, Belgium, and Switzerland. In Cologne, Stollwerck operates a fully automated chocolate packing line which achieves a packing rate of 1,000 chocolates per minute. The company achieves an annual turnover of approx. 500 million euros and produces an output of more than 100,000 tons of sweet products. Since 2011, Stollwerck has been part of the Baronie Group. www.stollwerck.de

Saison-Neuheit.
STOLLWERCK
Sprechende Chocolade-Tafel
(Qualität „Extra zart")
Singt musiziert deklamiert.
STOLLWERCK
ZUR AUFBEWAHRUNG VON 36 PHONOGRAPHEN-PLATTEN
I., Hoher Markt 6 (Ecke der Wipplingerstrasse)
GEBR. STOLLWERCK, WIEN
K. u. k. Hof-Schokolade-Fabrikanten.

A Stollwerck ad from the early 20th century read "Stollwerck's talking chocolate plate - sings, plays music, and recites," advertising these chocolate cylinders for phonographs as a "season's novelty"

Out Goes SAP and P2, in Comes Prodin-P3

After making the decision, Prodin set to work in February 2012 with a very ambitious time frame, to replace SAP before October 1. To illustrate just how hefty this project was, consider these key factors. The migration involved 300 SAP users in three countries and seven locations. Stollwerck processes about 120,000 sales orders per year, 75% of which are EDI-based. About 700,000 pallets of chocolates per year leave the production sites. Stollwerck delivers product to virtually every retailer in Europe, as well as customers in the U.S. and Australia, either directly from the plants or from one of the four distribution warehouses with 120,000 air conditioned warehouse location bins. An ambitious time frame, indeed.

It goes without saying that efficient project management was crucial for the project's success. Stollwerck commissioned Prodin to map the following functionalities in P3:

- Finance
- Sales
- EDI processes
- Quality assurance
- Purchase
- Material management
- Production
- Distribution processes, both to customers and intercompany
- Distributed warehouses

To complete the IT landscape, Prodin was to implement complex interfaces for third party software catering for sales forecasting, payroll, and Lotus Notes, which they use to manage mail workflow and product development. BASIS Europe stood by, supporting Prodin with the software installation and addressing specific questions about the best use of their BBj Data Server, Visual PRO/5, and BBj.

Ambitious Roadmap

The project roadmap entailed implementing the Stollwerck production sites first and creating temporary interfaces with other systems. The German sites in Berlin, Saalfeld, and Norderstedt went live in September, according to plan. The Norderstedt implementation was deliberately postponed in order to perform additional testing on the warehouse management storage strategy. The scanners in the warehouse connect with Prodin-P3 so that all WMS movements throughout the warehouse can be monitored in real-time.

On December 1, the Stollwerck sales organization, finance, and EDI with customers and shipping companies went live. They dedicated the year 2013 to bringing the sites in line and to optimize specific processes. In the second half of 2013, the implementation of the distribution warehouse in Belgium was finalized. In 2014, the Baronie sites in Belgium and the Netherlands will be brought up to speed – all within the planned time-frame, which not only included implementation of the software, but also the training of the users to keep them at the same productivity level throughout the migration process.

New Challenges

While the migration is under way in the various Baronie locations, Prodin is already working on an extension to Prodin-P3 that will allow all corporate account managers to enter, share, and distribute their customer sales forecasts. The new module will seamlessly integrate into the Prodin menu and be developed completely in BBj. It is scheduled to go live on January 1, 2014.

In June 2014, Stollwerck will switch their shipping partner, which implies another challenging project for Prodin. The distribution warehouse in Nohra near Weimar will move to a new site with 25,000 storage locations near Berlin, under management of the third party logistics supplier Fiege Logistik.

Summary

When chocolate maker Baronie Group took over Stollwerck in 2011, they decided to unify their IT landscape. They replaced SAP at Stollwerck with P3, Prodin's most recent ERP system release built on BBj and Visual PRO/5. Prodin managed the initial project and delivered it on time and in accordance with the customer's expectations. Once the migration is 100% complete, the whole Baronie group will profit from streamlined processes, mobile access to important company data, high data security standards and Prodin's comprehensive services. ■

Sweet Products/Baronie

Belgium-based Sweet Products, a privately owned company, is the parent of the Baronie Group of companies. Baronie produces a wide range of chocolate products for their international customer base. Baronie owns and operates three production sites in the Netherlands and Belgium, and an ultra-modern logistical center for warehousing and co-packed activities, located in Belgium. Baronie is a strategic supplier for virtually all major European retailers, for whom it provides branded products as well as an extensive range of private-label products. The company was founded in 1896 in the Netherlands. www.baronie.com

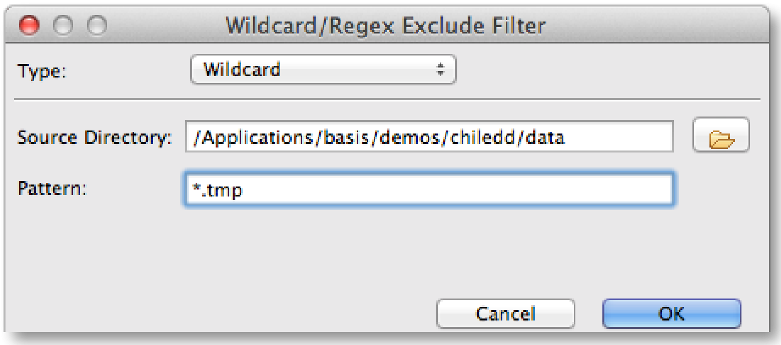
Sophisticated Regex Replication Enhancements

Replication provides BASIS DBMS application users the ability to replicate all or part of their database to one or more locations for the purpose of backup and/or data warehousing. A replication job essentially consists of a list of files and directories to include in the job and a list of files and directories to exclude specifically. While this generally covers the needs of most users, it is not uncommon for a user to need to include or exclude items that match particular filter criteria. In version 14.0, previewed in BBj® 13.10, replication users have the ability to add regular expression and glob filtering to the replication inclusion and exclusion lists.

By Jeff Ash
Software Engineer

Adding a Glob or Regular Expression Filter

Glob filters are the easiest to use and probably most familiar to users, e.g., wildcard filters like *.tmp, prefix*.*, etc. They provide a simple way to identify entire groups of files based on the name of the file. **Figure 1** shows a wildcard exclude filter that will instruct the replication job to ignore all files with the 'tmp' extension.



Wildcard/Regex Exclude Filter

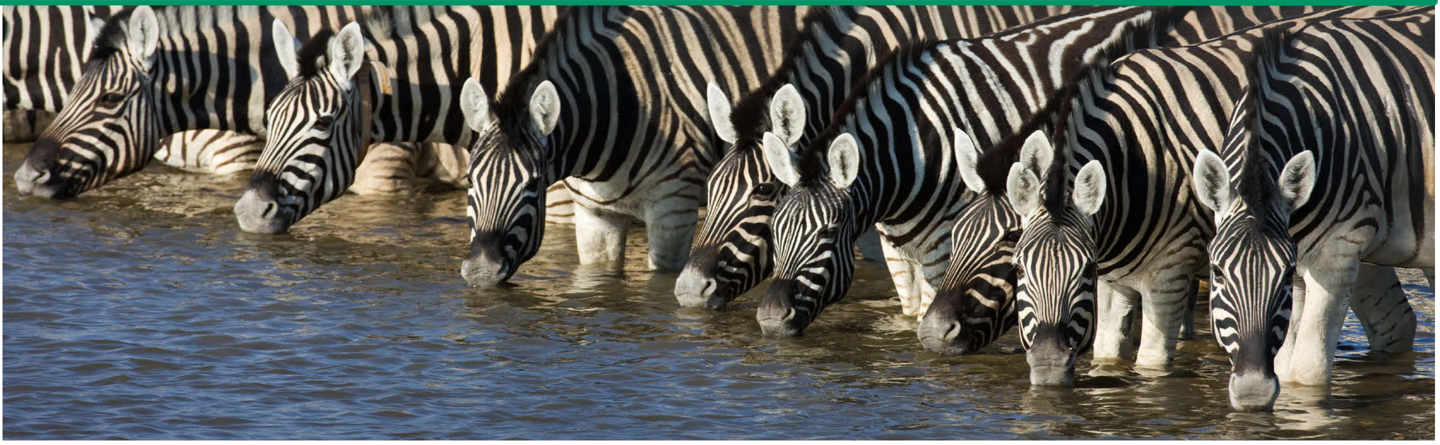
Type: Wildcard

Source Directory: /Applications/basis/demos/chiledd/data

Pattern: *.tmp

Figure 1. Specifying a wildcard exclusion

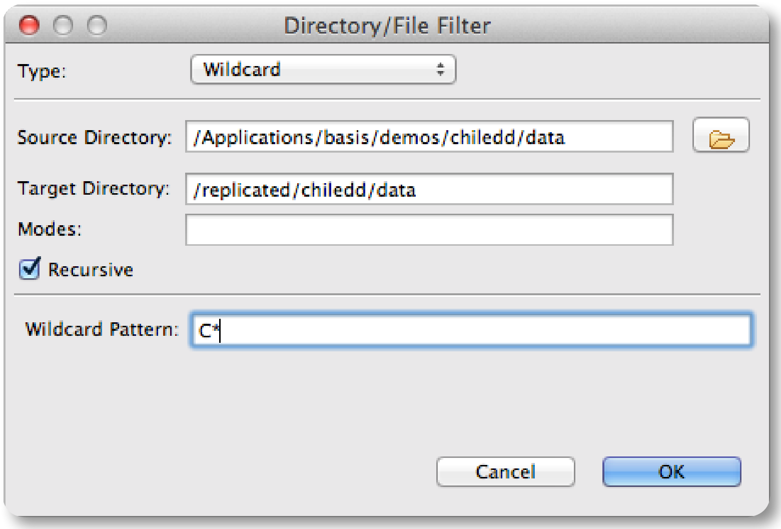
Regular expressions are also available as a filter type and while they can be more complex, they are also far more powerful and provide sophisticated searching and pattern matching that goes well beyond simple wildcards. Regular expressions allow for matching character classes and boundaries, alternation and grouping, repetition, and more. For more information on regular expressions, check out the Java Regular Expression Tutorial.



To add a filter, follow these simple steps:

1. Click the wildcard or regular expression button on the Replication Job Wizard on the appropriate page (included or excluded as appropriate) or while editing an existing replication job, on the appropriate included/excluded tab.
2. Select the directory where the filter applies.
3. Specify the glob or regular expression filter as appropriate.

Now, replication will validate file names with the filter before including or excluding them. An example of this appears in **Figure 2** that specifies a wildcard pattern of ‘C*’ for the types of files to include in the replication job.



Directory/File Filter

Type: Wildcard

Source Directory: /Applications/basis/demos/chiledd/data

Target Directory: /replicated/chiledd/data

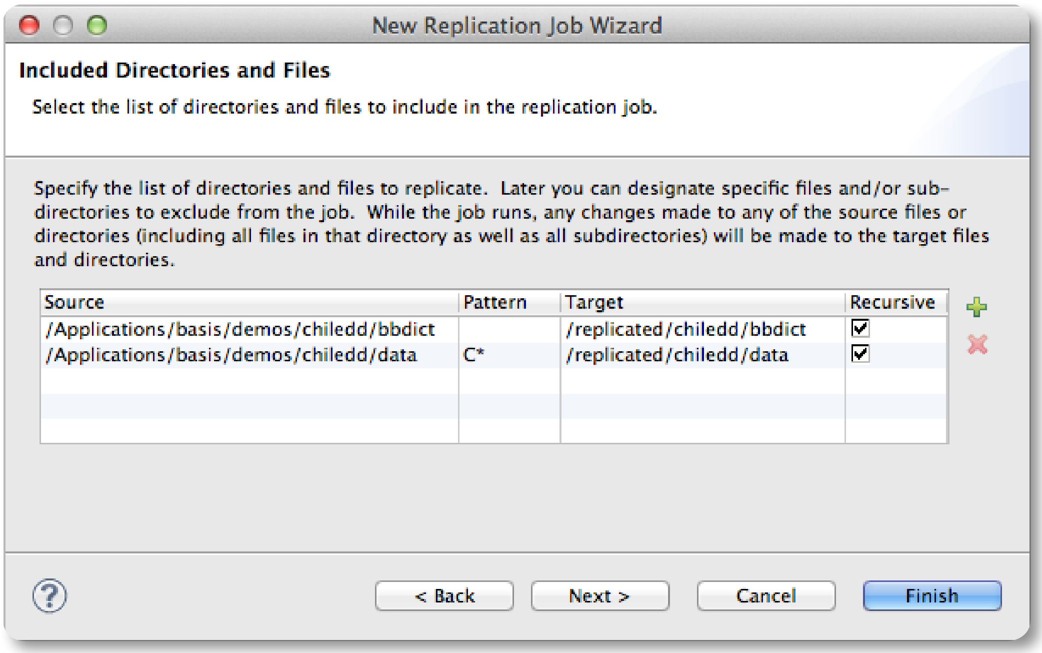
Modes:

☑ Recursive

Wildcard Pattern: C*

Figure 2. Specifying a wildcard inclusion

Figure 3 shows how this file filter appears after creation and how it is available in the list of directories and files to include in a new replication job.



New Replication Job Wizard

Included Directories and Files

Select the list of directories and files to include in the replication job.

Specify the list of directories and files to replicate. Later you can designate specific files and/or sub-directories to exclude from the job. While the job runs, any changes made to any of the source files or directories (including all files in that directory as well as all subdirectories) will be made to the target files and directories.

Source	Pattern	Target	Recursive
/Applications/basis/demos/chiledd/bbdict		/replicated/chiledd/bbdict	✓
/Applications/basis/demos/chiledd/data	C*	/replicated/chiledd/data	✓

Figure 3. Creating a new replication job with the newly defined directory/file filter

Summary

If you have ever spent any amount of time building a list of files to exclude from a replication job, such as temporary files generated by reporting or monthly closing processes, you have probably wondered if there was a better way to accomplish the task. With the introduction of powerful filtering support built into the replication interface, it is now possible to easily include or exclude entire groups of files based on naming conventions, even those that may not exist at job creation time. Whether you choose to use a simple glob filter, or pull out the big guns and use regular expressions for complete control, it is now a cinch to indicate categories of files to ignore or include. ■

For more information, read
- *Replication Introduction* in the online help at links.basis.com/replication
- *Anatomy of a Replication Job* at links.basis.com/11replication

Improve and Beautify Barista Apps With CSS

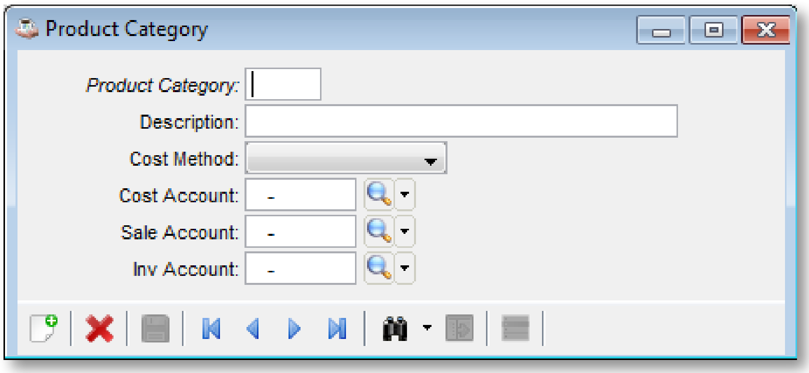
Since the dawn of graphical devices, the look and feel of an application has played an important role in communicating corporate image and acting as the common thread in product lines. Cascading style sheets (CSS) are the most popular means of applying visual and, to some extent, behavioral changes to web applications running in browsers without the need to make changes in the programs backing those applications.

BASIS Enhancements

BASIS extended BBj® and made it possible to run BBj programs in a browser user interface (BUI). The technology that BASIS implemented naturally includes the ability to apply CSS to BUI applications. The Barista® Application Framework is comprised of a set of BBj programs and as such, can run in BUI. Although developers could apply CSS at the BBj component level, one couldn't apply CSS to specific Barista forms and input controls...until now. In addition, BASIS now also extends to Barista the ability to set one or more CSS style names that apply visual effects at the form and column (field) levels.

Learn by Example

The Barista Examples Product Category alias "EXM_CATEGORY" is an example we'll look at to learn about assigning CSS style names and applying CSS to a Barista BUI form. When running in GUI, the form looks like **Figure 1**.



Product Category

Product Category:	
Description:	
Cost Method:	
Cost Account:	-
Sale Account:	-
Inv Account:	-

Figure 1. Original Product Category form in GUI

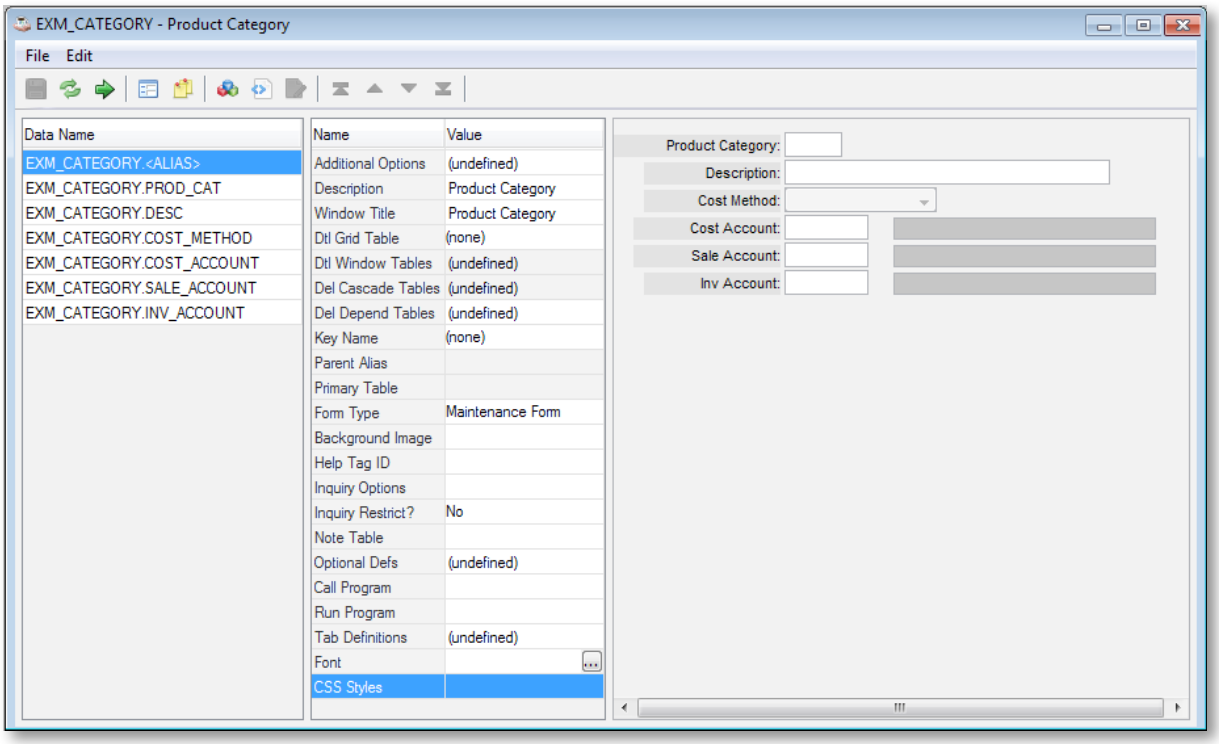
What we would like to do is preemptively give the user a visual indication of which fields, other than the key fields, require a value. For this example, we will make the "Description" column a required entry and set a property called "required" to its "CSS Styles" attribute. Secondly, we will configure Barista's default CSS file "stylesheet.css" to set the background color of the input control for the description column to yellow.

Although we won't use the "CSS Styles" attribute at the form level, I would like to briefly show you where to find it. To locate the form level attribute, double-click on a table alias name in the "Barista Development/Form Manager" menu to load the form

By Ralph Lance
Software Engineer

definition in the Form Designer. Once again, look at the Barista Examples Product Category alias "EXM_CATEGORY" in **Figure 2**.

After selecting the "EXM_CATEGORY.<ALIAS>" entry, notice a list of form attributes appears in the middle column. The new attribute, "CSS Styles," now appears highlighted at the bottom of the attribute list.



EXM_CATEGORY - Product Category

File Edit

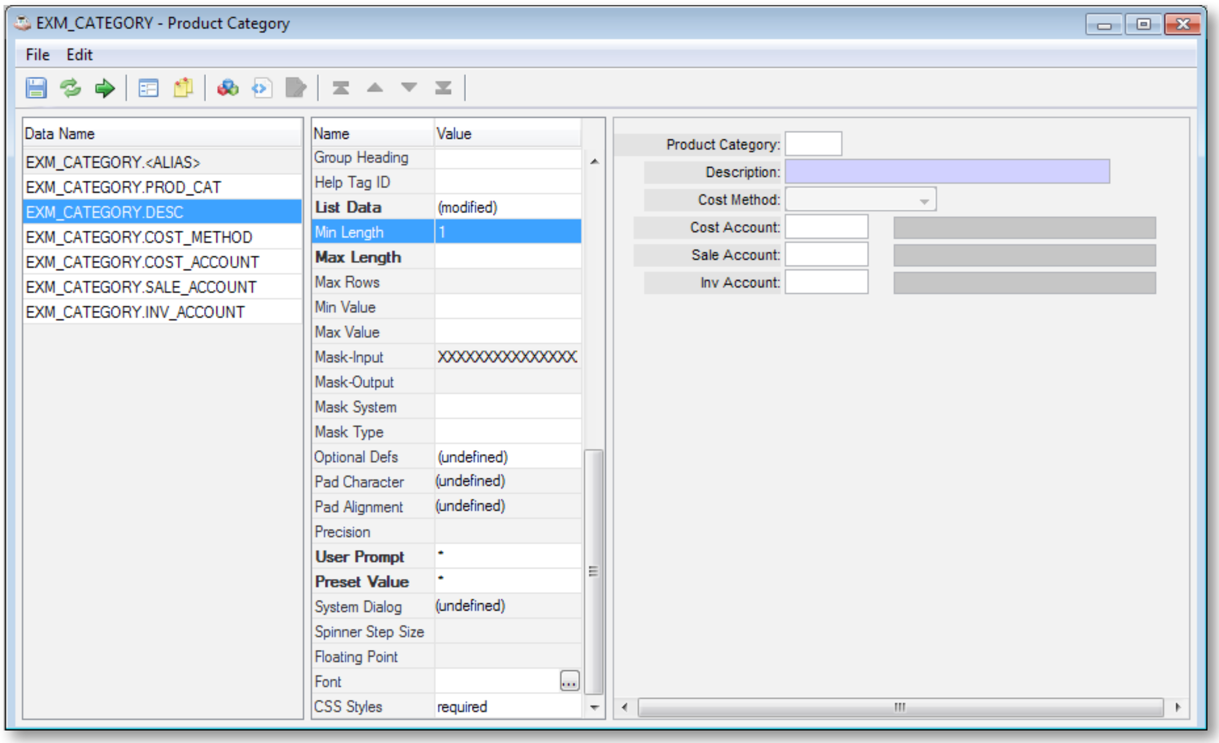
Data Name
EXM_CATEGORY.<ALIAS>
EXM_CATEGORY.PROD_CAT
EXM_CATEGORY.DESC
EXM_CATEGORY.COST_METHOD
EXM_CATEGORY.COST_ACCOUNT
EXM_CATEGORY.SALE_ACCOUNT
EXM_CATEGORY.INV_ACCOUNT

Name	Value
Additional Options	(undefined)
Description	Product Category
Window Title	Product Category
Dtl Grid Table	(none)
Dtl Window Tables	(undefined)
Del Cascade Tables	(undefined)
Del Depend Tables	(undefined)
Key Name	(none)
Parent Alias	
Primary Table	
Form Type	Maintenance Form
Background Image	
Help Tag ID	
Inquiry Options	
Inquiry Restrict?	No
Note Table	
Optional Defs	(undefined)
Call Program	
Run Program	
Tab Definitions	(undefined)
Font	
CSS Styles	

Product Category:
Description:
Cost Method:
Cost Account:
Sale Account:
Inv Account:

Figure 2. Sample Form Designer

Similar to the form alias, one can select a column in the Form Designer, either from the list of data names on the left or by clicking the field in the graphical frame. In order to set a column to be a required entry, set the minimum length attribute to 1, thus stating that at least one character must be entered in the field. We also assign the property name "required" as the value of the "CSS Styles" attribute. Note the use of the plural "Styles;" you can assign a comma-separated list of property names to a form or column. The result of our assignments appear in **Figure 3** where the "Min Length" attribute is set to "1" and the "CSS Styles" is now set to "required."



EXM_CATEGORY - Product Category

File Edit

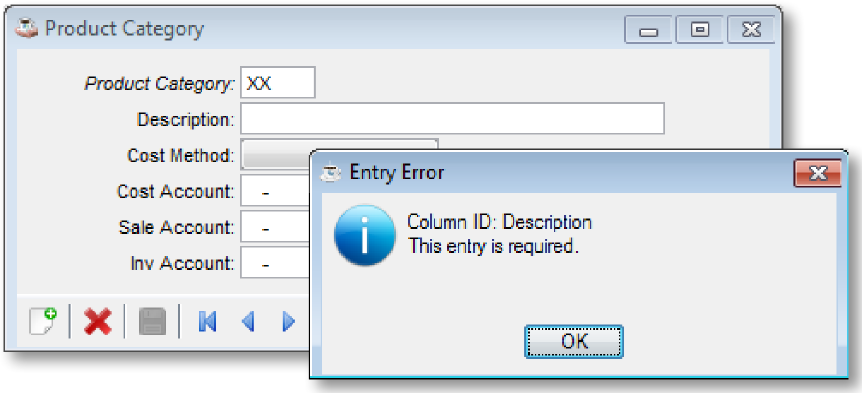
Data Name
EXM_CATEGORY.<ALIAS>
EXM_CATEGORY.PROD_CAT
EXM_CATEGORY.DESC
EXM_CATEGORY.COST_METHOD
EXM_CATEGORY.COST_ACCOUNT
EXM_CATEGORY.SALE_ACCOUNT
EXM_CATEGORY.INV_ACCOUNT

Name	Value
Group Heading	
Help Tag ID	
List Data	(modified)
Min Length	1
Max Length	
Max Rows	
Min Value	
Max Value	
Mask-Input	XXXXXXXXXXXXXXX
Mask-Output	
Mask System	
Mask Type	
Optional Defs	(undefined)
Pad Character	(undefined)
Pad Alignment	(undefined)
Precision	
User Prompt	*
Preset Value	*
System Dialog	(undefined)
Spinner Step Size	
Floating Point	
Font	
CSS Styles	required

Product Category:
Description:
Cost Method:
Cost Account:
Sale Account:
Inv Account:

Figure 3. Edited column attributes

Next, build the form and run it in GUI to show that the "Description" field is now required, as shown in **Figure 4**.


Product Category

Product Category: XX
Description:
Cost Method:
Cost Account: -
Sale Account: -
Inv Account: -

Entry Error

Column ID: Description
This entry is required.

OK

Figure 4. Entering no value in the Description field triggers an error

Convinced that everything is correct in Barista, we only need to define the CSS property "required" and run the Examples Product Category form in BUI.

Barista has its own CSS stylesheet file located under the Barista installation directory **`\sys\config\stylesheet.css`**, and uses this file for every Barista application that runs in BUI. We define the property "required" to set the background of any BBj control having this style name to yellow as follows:

```
.required {
    background-color: yellow;
    }
```

Running a form from the Barista MDI in BUI is as simple as right clicking on the menu item for the form and selecting [Launch in browser] from the context menu (**Figure 5**).

As we can see in **Figure 6**, the form displays using the CSS configuration, external to Barista, to give the user immediate visual feedback of which fields are required, thereby avoiding a potentially irritating error message for the user.

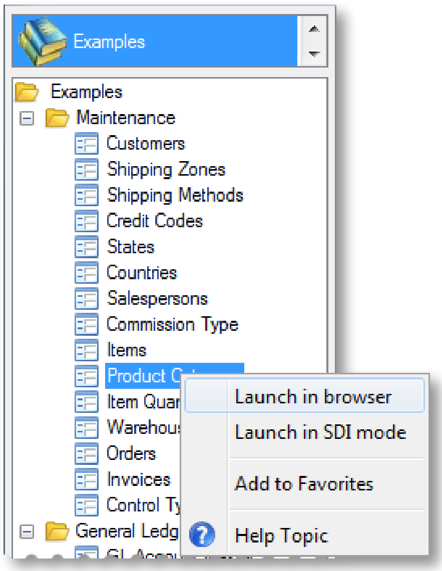
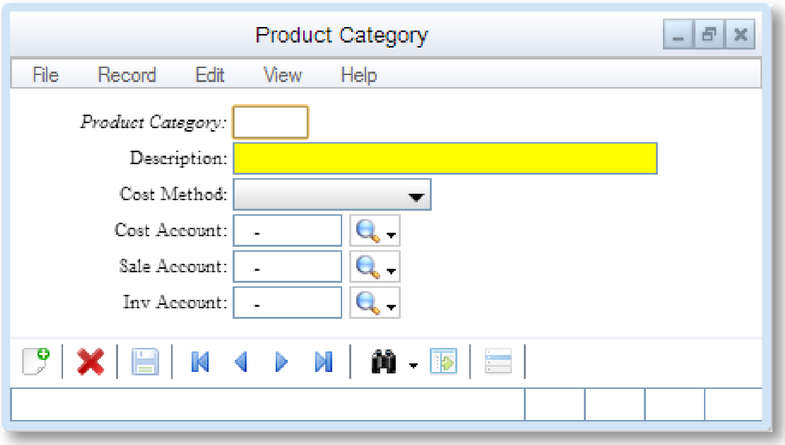


Figure 5. Launching a Barista form In BUI


Product Category

File Record Edit View Help

Product Category:
Description:
Cost Method:
Cost Account: -
Sale Account: -
Inv Account: -

Figure 6. Product Category form with CSS applied in BUI

Summary

Our example, albeit a simple one, is a real-world application of CSS. Now that you can assign CSS style names to Barista forms and fields, the door is wide open for developers to apply CSS to their own Barista applications. This capability allows developers to improve the usability of their applications and perhaps address the challenge of different corporate and product images among customers, without having to change the underlying code base. 🟥

- For more information, read *Adding Style to BBx Web Apps with Custom CSS* at links.basis.com/11css
- To learn more about BUI and applying CSS to your BUI applications, refer to
  - *BBj BUI: Getting Started* at links.basis.com/buiguide
  - *BUI: CSS API* at links.basis.com/cssapi

Revealing the Enhanced Data Dictionary

BBj® is a very powerful and flexible language, giving developers countless options for accessing and managing data. READ/WRITE RECORD provides the ability to control the appearance of the data down to the individual byte level. String templates add the ability to describe file record data as a set of fields of particular data types to make it easier to build and interpret the byte-oriented records used by the READ/WRITE RECORD calls. SQL goes a step further in providing a simple, high level, but very powerful querying language that makes it easier to acquire specific data and to gather that data from multiple sources into a single result. At the lowest level, all of these methods manipulate and access data in essentially the same fashion (with the exception of ESQL tables), writing and reading arrays of bytes to and from data files.

The Data Dictionary

A data dictionary is a layer that sits between the data file and the SQL engine. The data dictionary describes the entities within a particular database. Each dictionary describes a single database and includes information the SQL engine uses to perform queries and other database related operations on tables, views, stored procedures, sequences, user permissions, and roles (groups). In BBj 13.0 and higher, BASIS provides two dictionary formats to choose from: legacy and enhanced.

Looking Back at the Legacy Format

The legacy format is the dictionary format used by the SQL engine in PRO/5®, the BASIS ODBC Driver®, and earlier versions of BBj. The format has served its purpose well for many years providing BASIS developers with SQL access to their data files. However, this format has some limitations.

By Jeff Ash
Software Engineer

Four of the most notable examples are **1)** a 16-character limit on table, view, and column names, **2)** no way to define foreign key relationships (with the exception of ESQL tables), **3)** the need to specify a date format and date suffix to define columns as DATE type, and **4)** the inability to specify a specific precision and scale on NUMERIC type columns.

Introducing the Enhanced Data Dictionary

The enhanced format data dictionary overcomes the limitations of the legacy format. The following is a list of some of the key features in the enhanced format:

- **A more than 15-fold increase, to 255 characters, of the limit for entity names** – makes it possible to specify more descriptive table, view, and column names.
- **The ability to define foreign key relationships between the tables in the database** – information very useful for query building tools and to assist users in understanding the relationship between the tables in a database. It is important to note that this does not provide referential integrity (except on ESQL tables), but is simply informational.
- **Date formats defined at the column level** – embedded within the data dictionary so there is no need to configure the database with date format information. Another benefit is that column names do not need to adhere to any particular naming convention. For those who have multiple date formats used in their database, this makes it easy to accommodate that structure.
- **NUMERIC columns get precision and scale** – NUMERIC type columns contain a dictionary length that describes the length of the raw data at the file record level, but now additionally contain a precision and scale. This gives the database administrator the ability to describe the raw data, and exactly how the numeric values should look to the outside world. For example, a NUMERIC value in a BBj data file could be defined with a dictionary length of 12. But since BBj stores NUMERIC values as a string representation of that number, how would a third party application know how many decimal places were allowed? The answer is to provide precision and scale that allows a way to define this metadata.

Using the Enhanced Format

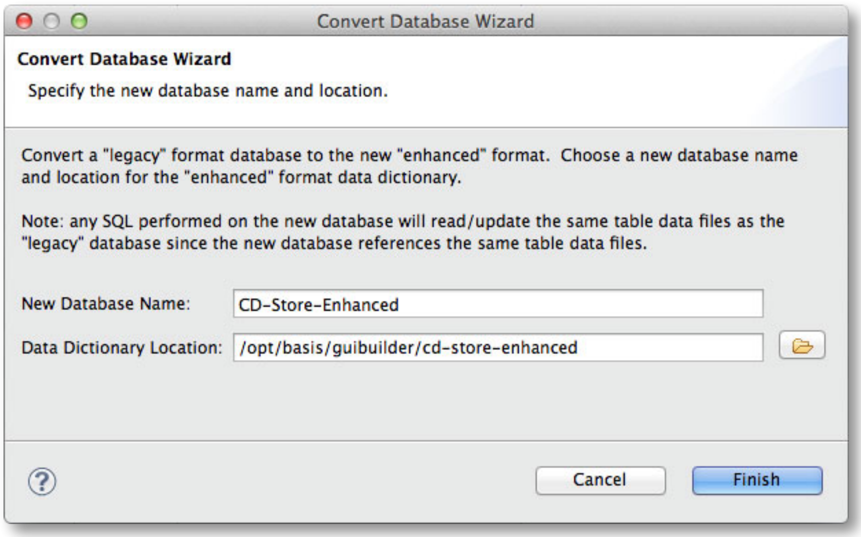
To begin using the enhanced format, create a new database from scratch, or convert an existing legacy format database to the enhanced format. Conversion is simple using the Enterprise Manager's database conversion wizard, shown in **Figure 1** (you can also run a conversion using the Admin API that is outside of the scope of this article). Converting your existing data dictionary to an enhanced data dictionary is as simple as using the following steps in the database conversion wizard:

1. Log into the Enterprise Manager.
2. Double-click on the Databases node.
3. Select the database to convert.
4. Click the "Convert to Enhanced" button.
5. Specify a name for the new database.
6. Specify a location for the new data dictionary.

The conversion creates a new data dictionary in a new location, while preserving your original data dictionary. (it does not modify or overwrite your legacy format dictionary). It will set the date format on all columns that match any date format configuration settings on the legacy database and store these in the enhanced format column definitions automatically. This conversion process creates the new database and points it to the existing table data files. As a result, you can continue to use the legacy database while testing the new enhanced format database. Just keep in mind any UPDATE, DELETE or INSERT statements will modify the table data for both database definitions since they both use the same table data files.

Exploring the features of the enhanced format dictionary is as easy as double-clicking on the enhanced format database, then selecting the Tables tab. Double-click on a table to open up the table editor where you can set the precision and scale, add foreign key references, and specify date formats for specific columns. **Figure 2** shows Enterprise Manager's foreign key management user interface for an enhanced version of the Chile Company database. This screen lists the currently defined foreign keys and allows developers to add, remove, and modify existing defined relationships between tables.

After defining foreign key relationships between the ORDER_LINE table and the ORDER_HEADER and ITEM tables on their common columns, we can view the relationships in a


Convert Database Wizard

Convert Database Wizard

Specify the new database name and location.

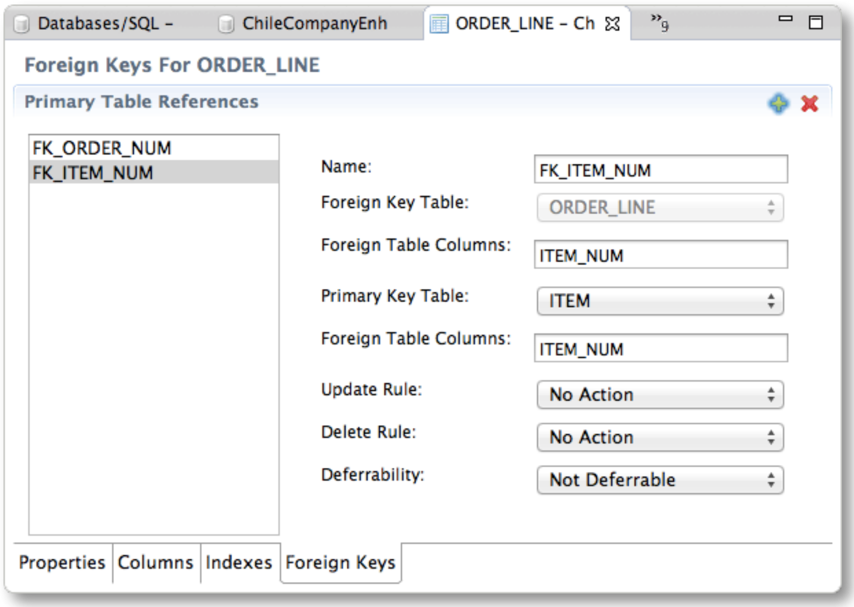
Convert a "legacy" format database to the new "enhanced" format. Choose a new database name and location for the "enhanced" format data dictionary.

Note: any SQL performed on the new database will read/update the same table data files as the "legacy" database since the new database references the same table data files.

New Database Name: CD-Store-Enhanced

Data Dictionary Location: /opt/basis/guibuilder/cd-store-enhanced

Figure 1. The database conversion wizard



Foreign Keys For ORDER_LINE

Primary Table References

FK_ORDER_NUM
FK_ITEM_NUM

Name: FK_ITEM_NUM

Foreign Key Table: ORDER_LINE

Foreign Table Columns: ITEM_NUM

Primary Key Table: ITEM

Foreign Table Columns: ITEM_NUM

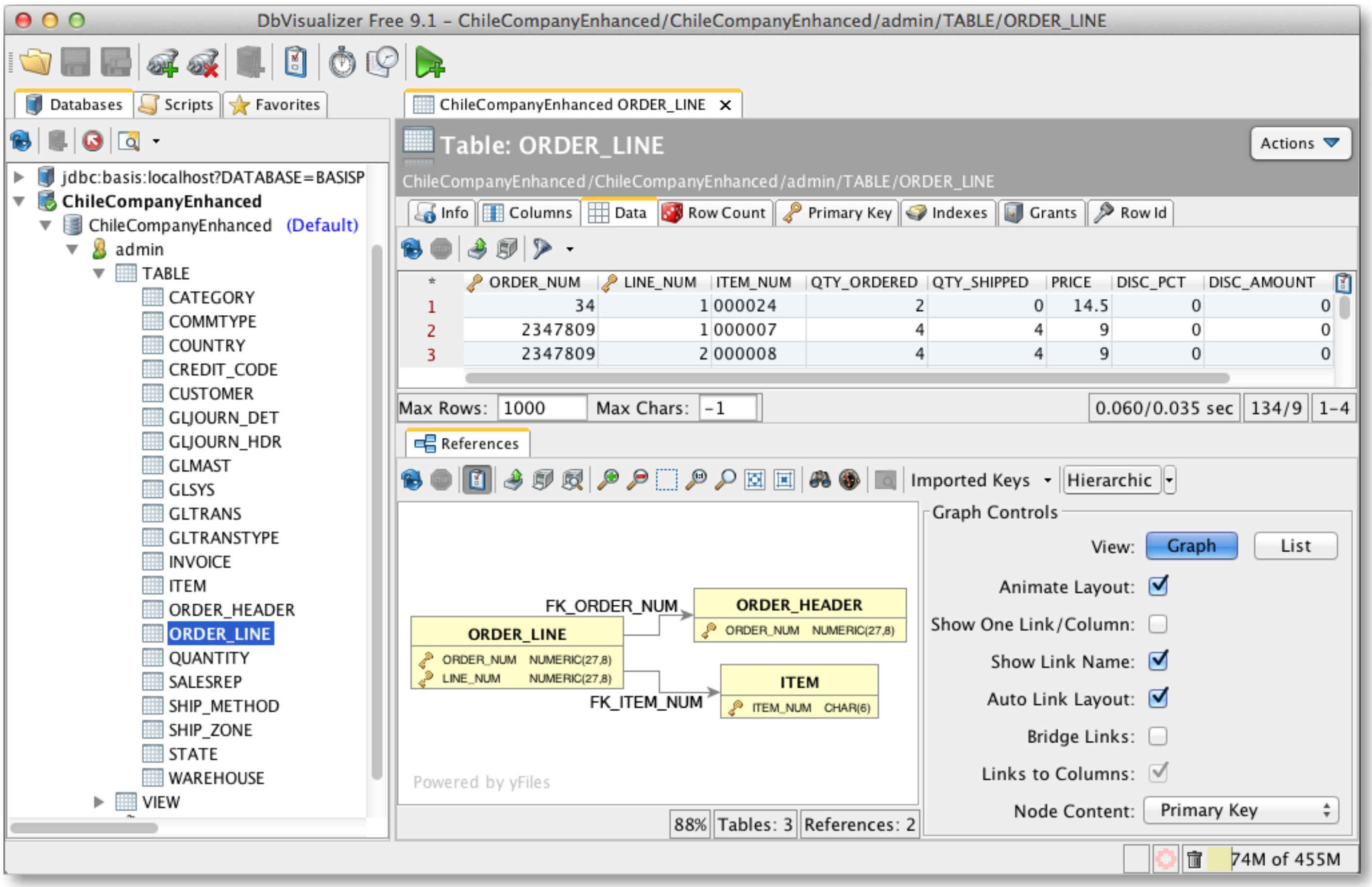
Update Rule: No Action

Delete Rule: No Action

Deferrability: Not Deferrable

Figure 2. Enterprise Manager's foreign key management

third party tool, such as DbVisualizer (www.dbvis.com). **Figure 3** shows a screenshot of DbVisualizer examining the ORDER_LINE table, displaying its data and a graphical representation of the foreign key relationships that we just created.



DbVisualizer Free 9.1 – ChileCompanyEnhanced/ChileCompanyEnhanced/admin/TABLE/ORDER_LINE

Databases | Scripts | Favorites

- jdbc:basis:localhost?DATABASE=BASISP
- **ChileCompanyEnhanced**
  - ChileCompanyEnhanced (Default)
    - admin
      - TABLE
        - CATEGORY
        - COMMTYPE
        - COUNTRY
        - CREDIT_CODE
        - CUSTOMER
        - GLJOURN_DET
        - GLJOURN_HDR
        - GLMAST
        - GLSYS
        - GLTRANS
        - GLTRANSTYPE
        - INVOICE
        - ITEM
        - ORDER_HEADER
        - ORDER_LINE
        - QUANTITY
        - SALESREP
        - SHIP_METHOD
        - SHIP_ZONE
        - STATE
        - WAREHOUSE
      - VIEW

ChileCompanyEnhanced ORDER_LINE

Table: ORDER_LINE

ChileCompanyEnhanced/ChileCompanyEnhanced/admin/TABLE/ORDER_LINE

Info | Columns | Data | Row Count | Primary Key | Indexes | Grants | Row Id

*	ORDER_NUM	LINE_NUM	ITEM_NUM	QTY_ORDERED	QTY_SHIPPED	PRICE	DISC_PCT	DISC_AMOUNT
1	34	1	000024	2	0	14.5	0	0
2	2347809	1	000007	4	4	9	0	0
3	2347809	2	000008	4	4	9	0	0

Max Rows: 1000 Max Chars: -1 0.060/0.035 sec 134/9 1-4

References

Imported Keys Hierarchic

Graph Controls

View: Graph List
Animate Layout: ✓
Show One Link/Column: ☐
Show Link Name: ✓
Auto Link Layout: ✓
Bridge Links: ☐
Links to Columns: ✓
Node Content: Primary Key

88% Tables: 3 References: 2

74M of 455M

Figure 3. DbVisualizer showing the foreign key references for the ORDER_LINE table

Summary

BBj 13.0 introduced a new data dictionary format to expand the capabilities of the BBj database. These new features provide better interoperability with third party applications with the addition of foreign key relationships and precision and scale, increased flexibility with the removal of name limits, and simplified database configuration by storing date format information in the column definitions. This suite of new features makes the new enhanced format data dictionary the choice for new database projects, and something to seriously consider with existing legacy applications as it further empowers third party access to existing BBj data. The biggest benefit of converting your existing data dictionary today, is that your data dictionary will now be ready to handle any and all of the new features that your organization will need in the future, but will continue to work exactly like the original data dictionary until you begin to utilize the new features and functions. ■

Take a 30 minute break to self-serve a

links.basis.com/javabreak

Enterprise Administration To-Go

In August 2013, Statista reported that 17.4% of global web traffic came through mobile devices while some continents saw over 25%. This volume of mobile device usage to carry out Internet-based daily tasks is only expected to grow. BASIS sits at the forefront of this trend providing its developers the ability to deploy their standard desktop BBj® applications as an entirely browser-based application with little or no code modification using BASIS' unique browser user interface (BUI) technology. In parallel, BBj version 13.0 takes a significant step forward to provide two new powerful flavors of our Java-based administration tool; a plug-in to the BASIS Eclipse IDE and a new browser version suitable for both mobile and desktop use.

By Jeff Ash
Software Engineer

The New Enterprise Manager

BBj is a powerful application platform requiring numerous configuration capabilities including basic server settings, database management, scheduling tasks, and managing replication, just to name a few. The Enterprise Manager, which provides this administration capability, is all-new in BBj 13.0. Leveraging the power and extensible nature of the Eclipse platform, BASIS now offers the Enterprise Manager as a part of the BASIS

Eclipse IDE as well as a web-based version that utilizes the same interface in both versions. Let's take a look at the new Enterprise Manager.

Browser-based

The most prominent feature of the new Enterprise Manager (EM) its browser-based version – the Browser EM – that requires no Java on the client machine and no need for installing any plug-ins or applications. To access the Browser EM, users simply point their browser to the location of the web server that is serving up the Browser EM web application. With a default installation of BBj Services, the built-in Jetty Web Server provides access at `localhost:8888/bbjem/em`. Users can then access the Browser EM from modern browsers in desktops and mobile devices such as iOS, Android, Windows, and BlackBerry powered tablets (**Figure 1**) and smartphones.

Multiple Configuration Tabs (Eclipse GUI Version)

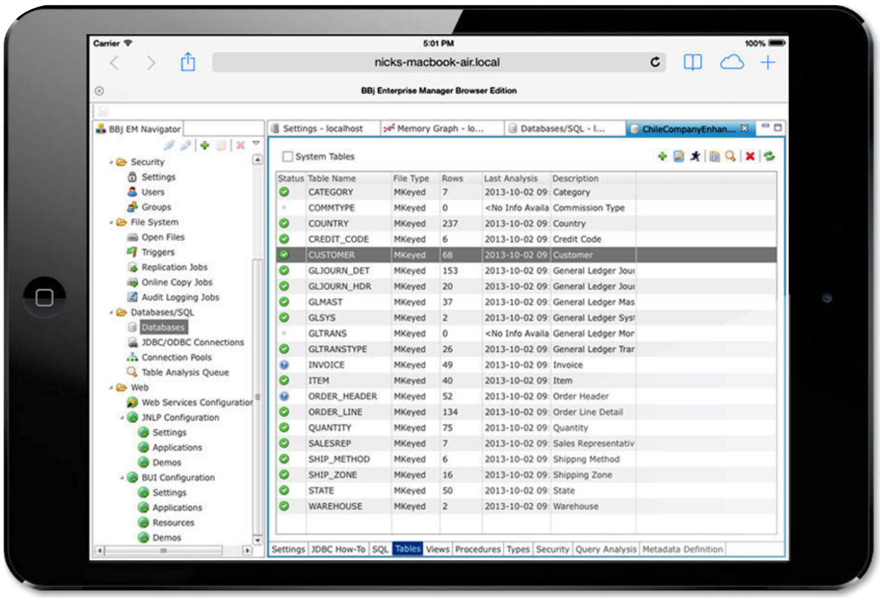
Eclipse provides a nice multi-tab editor interface so administrators can now open multiple configuration items at the same time. In the Eclipse version, they can arrange these tabs in such a way as to view multiple tabs at one time, as shown in **Figure 2**.

Log File Viewer With Incremental Search

The new EM boasts an improved log file viewer that shows a list of the available log files as well as last modified information and the log file size. Administrators can open and view multiple log files at the same time. Furthermore, the interface provides the ability to perform an incremental search and filter out any lines in the log file that do not match the filter criteria. **Figure 3** is an example of EM filtering a log file to display only lines that contain the **'cust_num'** value.

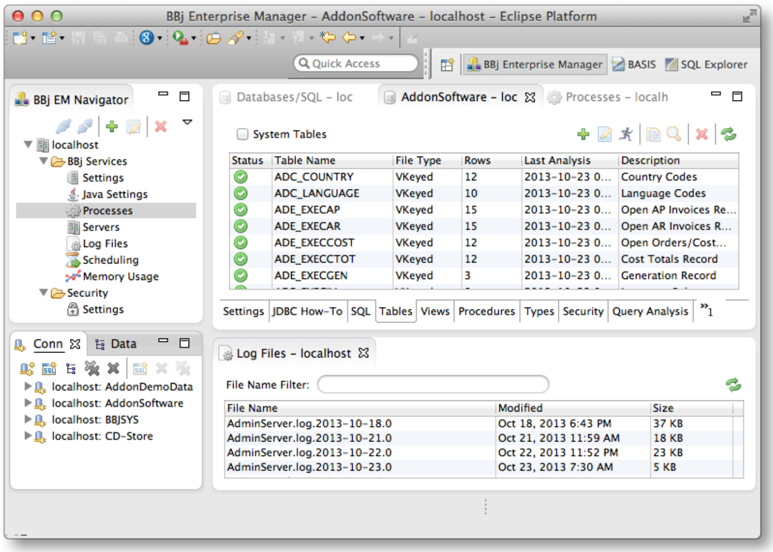
Filtering

Filtering is a convenient new feature in both the Eclipse and Browser EM that greatly simplifies the process of locating specific information within a large set of open files, processes, database connections, etc. Expand the "More Options" disclosure icon at the top of a list (if available) to see the filtering options. Filtering is easy using a dropdown to select the field to filter, the



Status	Table Name	File Type	Rows	Last Analysis	Description
✓	CATEGORY	MKeyed	7	2013-10-02 09:	Category
	COMMTYPE	MKeyed	0	<No Info Availa	Commission Type
✓	COUNTRY	MKeyed	237	2013-10-02 09:	Country
✓	CREDIT_CODE	MKeyed	6	2013-10-02 09:	Credit Code
✓	CUSTOMER	MKeyed	68	2013-10-02 09:	Customer
✓	GLJOURN_DET	MKeyed	153	2013-10-02 09:	General Ledger Jour
✓	GLJOURN_HDR	MKeyed	20	2013-10-02 09:	General Ledger Jour
✓	GLMAST	MKeyed	37	2013-10-02 09:	General Ledger Mas
✓	GLSYS	MKeyed	2	2013-10-02 09:	General Ledger Sys
	GLTRANS	MKeyed	0	<No Info Availa	General Ledger Mor
✓	GLTRANSTYPE	MKeyed	26	2013-10-02 09:	General Ledger Trar
●	INVOICE	MKeyed	49	2013-10-02 09:	Invoice
✓	ITEM	MKeyed	40	2013-10-02 09:	Item
●	ORDER_HEADER	MKeyed	52	2013-10-02 09:	Order Header
✓	ORDER_LINE	MKeyed	134	2013-10-02 09:	Order Line Detail
✓	QUANTITY	MKeyed	75	2013-10-02 09:	Quantity
✓	SALESREP	MKeyed	7	2013-10-02 09:	Sales Representativ
✓	SHIP_METHOD	MKeyed	6	2013-10-02 09:	Shippng Method
✓	SHIP_ZONE	MKeyed	16	2013-10-02 09:	Shipping Zone
✓	STATE	MKeyed	50	2013-10-02 09:	State
✓	WAREHOUSE	MKeyed	2	2013-10-02 09:	Warehouse

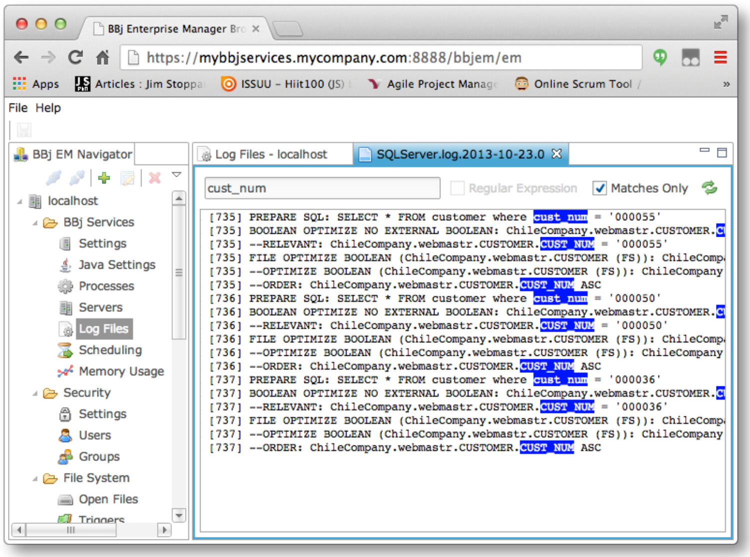
Figure 1. Enterprise Manager running on an iPad mini in Mobile Safari



Status	Table Name	File Type	Rows	Last Analysis	Description
✓	ADC_COUNTRY	VKeyed	12	2013-10-23 0...	Country Codes
✓	ADC_LANGUAGE	VKeyed	10	2013-10-23 0...	Language Codes
✓	ADE_EXECAP	VKeyed	15	2013-10-23 0...	Open AP Invoices Re...
✓	ADE_EXECAR	VKeyed	15	2013-10-23 0...	Open AR Invoices R...
✓	ADE_EXECCOST	VKeyed	12	2013-10-23 0...	Open Orders/Cost...
✓	ADE_EXECCTOT	VKeyed	12	2013-10-23 0...	Cost Totals Record
✓	ADE_EXECGEN	VKeyed	3	2013-10-23 0...	Generation Record

File Name	Modified	Size
AdminServer.log.2013-10-18.0	Oct 18, 2013 6:43 PM	37 KB
AdminServer.log.2013-10-21.0	Oct 21, 2013 11:59 AM	18 KB
AdminServer.log.2013-10-22.0	Oct 22, 2013 11:52 PM	23 KB
AdminServer.log.2013-10-23.0	Oct 23, 2013 7:30 AM	5 KB

Figure 2. Displaying multiple tabs on both the left and right side of the interface



```
[735] PREPARE SQL: SELECT * FROM customer where cust_num = '000055'
[735] BOOLEAN OPTIMIZE NO EXTERNAL BOOLEAN: ChileCompany.webmastr.CUSTOMER.C
[735] --RELEVANT: ChileCompany.webmastr.CUSTOMER.CUST_NUM = '000055'
[735] FILE OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileComp
[735] --OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileCompany
[735] --ORDER: ChileCompany.webmastr.CUSTOMER.CUST_NUM ASC
[736] PREPARE SQL: SELECT * FROM customer where cust_num = '000050'
[736] BOOLEAN OPTIMIZE NO EXTERNAL BOOLEAN: ChileCompany.webmastr.CUSTOMER.C
[736] --RELEVANT: ChileCompany.webmastr.CUSTOMER.CUST_NUM = '000050'
[736] FILE OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileComp
[736] --OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileCompany
[736] --ORDER: ChileCompany.webmastr.CUSTOMER.CUST_NUM ASC
[737] PREPARE SQL: SELECT * FROM customer where cust_num = '000036'
[737] BOOLEAN OPTIMIZE NO EXTERNAL BOOLEAN: ChileCompany.webmastr.CUSTOMER.C
[737] --RELEVANT: ChileCompany.webmastr.CUSTOMER.CUST_NUM = '000036'
[737] FILE OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileComp
[737] --OPTIMIZE BOOLEAN (ChileCompany.webmastr.CUSTOMER (FS)): ChileCompany
[737] --ORDER: ChileCompany.webmastr.CUSTOMER.CUST_NUM ASC
```

Figure 3. Filtering the contents of the SQLServer log file in a browser

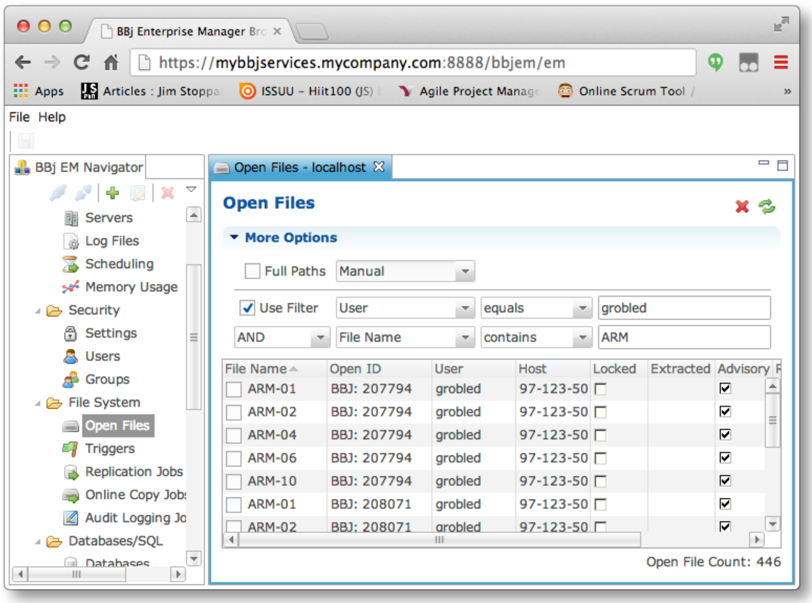
operator for the comparison such as equals, greater than, less than, contains, and even regex (regular expression matching); and a logical AND/OR for multiple filter criteria. **Figure 4** illustrates this capability with a filtered the list of open files that show only those files whose names contain the string ARM and are opened by the user grobled.

SQL Explorer (Eclipse GUI Version)

Included with the EM is the SQL Explorer plug-in. This Eclipse plug-in provides a powerful interface for building and executing queries. Some of the features include unlimited results, syntax color coding, executing multiple semi-colon delimited SQL statements, query history, database introspection, ability to connect to third party databases, and much more. **Figure 5** shows EM's capability to execute multiple SQL statements and display the results in a separate tab.

Summary

The features highlighted in this article are just a few of the capabilities in the new BBj Enterprise Manager. Administration is now possible using the BASIS Eclipse IDE or from any compatible web browser, whether on a remote PC, tablet, or even a smartphone. As BASIS continues to grow this exciting new component of its suite of tools, developers can expect to see more and more features to make the job of administering all aspects of BBj, both from the office or on the go, even easier. All of the features in this tool are made possible through a published and documented API, so that the community may utilize any of these features in their own applications and custom written utilities that they desire. ■



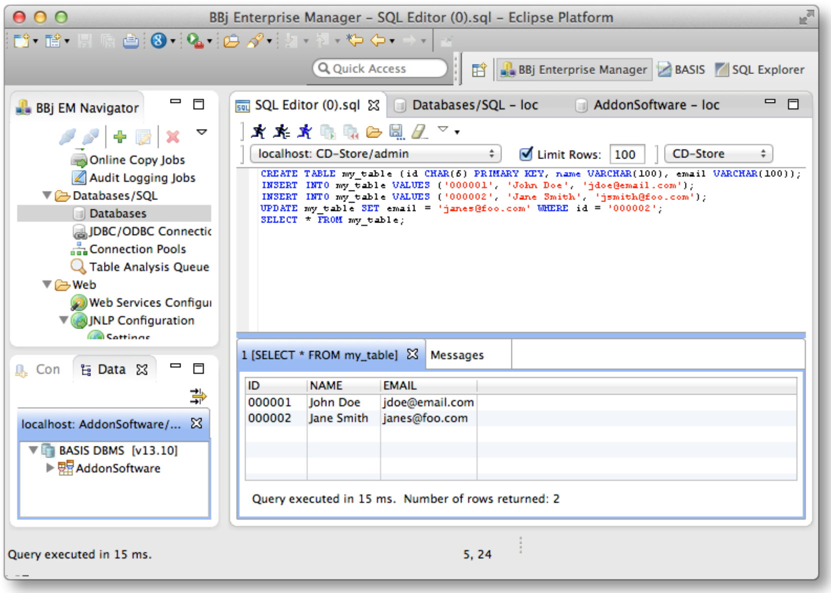
Open Files

Full Paths: Manual
☑ Use Filter: User equals grobled
AND File Name contains ARM

File Name	Open ID	User	Host	Locked	Extracted	Advisory
ARM-01	BBJ: 207794	grobled	97-123-50	☐		☑
ARM-02	BBJ: 207794	grobled	97-123-50	☐		☑
ARM-04	BBJ: 207794	grobled	97-123-50	☐		☑
ARM-06	BBJ: 207794	grobled	97-123-50	☐		☑
ARM-10	BBJ: 207794	grobled	97-123-50	☐		☑
ARM-01	BBJ: 208071	grobled	97-123-50	☐		☑
ARM-02	BBJ: 208071	grobled	97-123-50	☐		☑

Open File Count: 446

Figure 4. Filtering the list of open files for a particular user and partial filename



```
CREATE TABLE my_table (id CHAR(6) PRIMARY KEY, name VARCHAR(100), email VARCHAR(100));
INSERT INTO my_table VALUES ('000001', 'John Doe', 'jdoe@email.com');
INSERT INTO my_table VALUES ('000002', 'Jane Smith', 'jsmith@foo.com');
UPDATE my_table SET email = 'janes@foo.com' WHERE id = '000002';
SELECT * FROM my_table;
```

ID	NAME	EMAIL
000001	John Doe	jdoe@email.com
000002	Jane Smith	janes@foo.com

Query executed in 15 ms. Number of rows returned: 2

Query executed in 15 ms. 5, 24

Figure 5. Executing multiple SQL statements in a row and displaying the results

Sit back and enjoy a 30-minute presentation with BASIS!

links.basis.com/javabreak

BASIS has Git Bug Control, Get Git too!

You have probably experienced this dilemma at least once in your career. Somewhere along the way, a well-intentioned person made a change to your product's code that introduced a nasty bug. At the time, you didn't discover it because the code compiled just fine and everything seemed to run okay. Then on a dark and dreary day, a customer crashed into this rather major bug and promptly reported it to you. The customer discovered the bug long after you had made extensive changes to the code. Rather than keeping those changes and trying to patch them, it would be ideal if there was some way to just back the bug out of the code.

But how do you back a bug out? Is there a simple way?

If your code is just a set of files stored in a single location on the server, you have no choice but to try to patch the bug. If you keep backups, you can restore the file from a previous point in time, as long as your backups go back far enough and you know when someone made the offending change...exactly why developers invented source control. Source control systems keep backups of your precious files with a record of when changes were made, what files were changed, how the file content was changed, and who changed them. This makes recovery from this scenario possible and in most cases, very easy.

Source code control has been around in some form since the 1960's. There are a lot of choices out on the market today including Subversion, Microsoft Team Foundation Server, Mercurial, and Git, to name a few. If you work with any kind of files containing valuable information that changes over time, you will find source control invaluable since it allows you to track your changes and revert if necessary. This article takes a close look at how Git benefits BASIS and what it can do for you.

By Shaun Haney
Quality Assurance Engineer

Git, a Distributed Source Control System

Linus Torvalds, the chief architect of the Linux kernel, developed Git out of necessity to maintain the Linux kernel in a source control system. The nature of the Linux kernel project differs from a lot of single-company commercial projects, as its source is developed by various highly distributed contributors all over the world simultaneously.

To accommodate this extreme development environment, Torvalds developed Git as a distributed version control system, able to quickly process and compress large volumes of code, with support for parallel code development. The way this works is that people who want to contribute source to the archive get their own full copy of the archive. The archive is relatively small compared to the amount of code it actually holds. Each version or "commit" in the archive is just a collection of differences or "deltas" from the previous version. Users can update their individual archives from another archive, usually the original archive, by performing a "pull" from that archive. Likewise, once users have changes they would like to contribute to another archive, they can "push" their changes to that archive.

If users are working on long-term changes, they can create their own "branch." Creating a separate branch allows a user to safely create and test changes without affecting the master branch. Once a user is sure that his changes are safe to incorporate into the master branch, he can merge in those changes.

Because of these features, Git has proven a valuable source control solution for the Linux kernel project. In the spirit of open source, Git is not only available to Linux kernel contributors, but to anyone who needs a distributed versioning system.

Git for Version Control

For BASIS, Git is an extremely valuable tool for source control for the AddonSoftware® project. This past June, BASIS converted the long-standing AddonSoftware Subversion (SVN) archive to Git because its development spans several companies with valuable contributions from multiple VARs. In addition, Git supports local version control and switching between branches, both of which satisfy critical needs in this environment. Developers get their own full copy of the repository and create a branch for the feature they add. As they continue to develop, they update regularly from BASIS' repository and push their feature back to BASIS' repository when it is ready. While SVN supports branching, Git puts the entire repository on the developer's local machine to allow quick and seamless switching between branches using the same working directory. This avoids having separate directories for each branch that require individual updating.

Git for Preserving Customizations Through the Upgrade Cycle

Git also serves as a valuable tool to help AddonSoftware partners preserve and update their vertical applications or customizations through the upgrade cycle. BASIS provides a read-only central Git repository that contains each major AddonSoftware release. When AddonSoftware partners want to upgrade their customers' installations with the new release, they clone the BASIS repository and roll it back to the same version that their customer has. Next, the partners add their verticals or customizations to the repository, then pull all the revisions, including the very latest release from the central BASIS archive. During the pull, a merging process takes place that updates the partner's verticals or customizations where needed. They can then port the upgraded vertical or customized version back to their customer's AddonSoftware installation and continue to reuse the cloned archive as the starting point for future upgrades. In this way, Git allows AddonSoftware partners to maintain their vertical or customizations through upgrades, making the overall upgrade process dramatically easier and, more importantly, affordable to the customer.

Imagine how much easier it would be to keep your vertical current with the annual upgrades from the original vendor if your source code control system could automatically manage the upgrades after your first integration. That is what Git does for you; you never have to struggle through an upgrade again and your customers can benefit from every new upgrade and security release with ease.

Git for You

In addition to Git's capabilities for large multi-party projects, Git certainly benefits smaller companies and individuals as well. Benefits include basic source control, speedy updates, local development, branching and merging support, and integrated support in the Eclipse IDE (links.basis.com/eclipse).

- **Basic source code control.** As mentioned in the introduction, any non-trivial project benefits highly from source code control. Source code control gives the developer the ability to back out changes or at least identify a specific change when product development has gone awry. The more frequently you check in your changes, the more you benefit. Most modern source code control systems offer branching and merging, allowing you to introduce drastic changes to your product that temporarily break everything for you without affecting anyone else so long as you're checking it into your own branch.
- **Speedy updates.** Git is a very fast, efficient system. Because of the way Git stores differences, you can often clone an entire repository in Git in as much time as it takes to check out a single revision in SVN. Even better, after you first clone the archive, any updates to or from another archive only consist of differences, making the transaction speedy.
- **Local development.** Unlike many other versioning systems, you do not need access to your central repository to check in changes to Git. You can continue to check in changes to your local repository and push those changes to the central repository once you have network access again.
- **Branching and merging.** Modern versioning systems support branching and merging, but Git excels at it. Since the entire repository is available to you locally, you can seamlessly jump between or create new branches as needed. Whenever you switch between branches, the checkout of the branch is always in your working directory. In centralized versioning systems like SVN, you need to check out the desired branches from the central repository, keeping each branch in its own directory. With Git, creating your own branch is easy and highly recommended if the change involves multiple files or blocks of code. Once you're ready to merge your changes in, Git remembers where you branched and tracks the changes accordingly. In fact, if you change a line of code in your branch, and someone else makes a different change in the same line in the source branch from which you created yours, Git notes the conflict and prevents these changes from overwriting each other.
- **Support in the Eclipse IDE.** BASIS has developed Eclipse plug-ins, introduced at TechCon2013 – the BBj Enterprise Manager and BASIS Development Tools. The Eclipse IDE also comes with EGit that provides Git integration with Eclipse for easy check-in or updating of the source within the IDE when working with BBj code.

Summary

Git is more than a versioning tool; it's a robust tool for anyone who needs to keep their own sets of code or to merge files. BASIS developers and AddonSoftware partners find Git to be a valuable multi-purpose tool with strong capabilities that dramatically improves their productivity.

Whether or not you use Git as your ultimate source code control tool, version control is an asset in all development environments, from the single developer to large, distributed networks. With all of the excellent choices available, everyone can find the tool that best suits their needs and begin reaping the benefits of version control. Once you make the transition, you will wonder how you ever lived without it! ■

Read *'Git'dy Up Developers!* at links.basis.com/12git

"Git" Started Now

Keeping your code under versioning control with Git sounds like a great idea. What do you need in order to get the process going?

- The Eclipse IDE comes with EGit, which has the tools you will need in order to work with Git repositories. Find information on how to get Eclipse and our plug-ins at links.basis.com/eclipse
- Learn how to use Git in the IDE, at wiki.eclipse.org/EGit/User_Guide
- Apress provides an excellent book on Git at git-scm.com/book

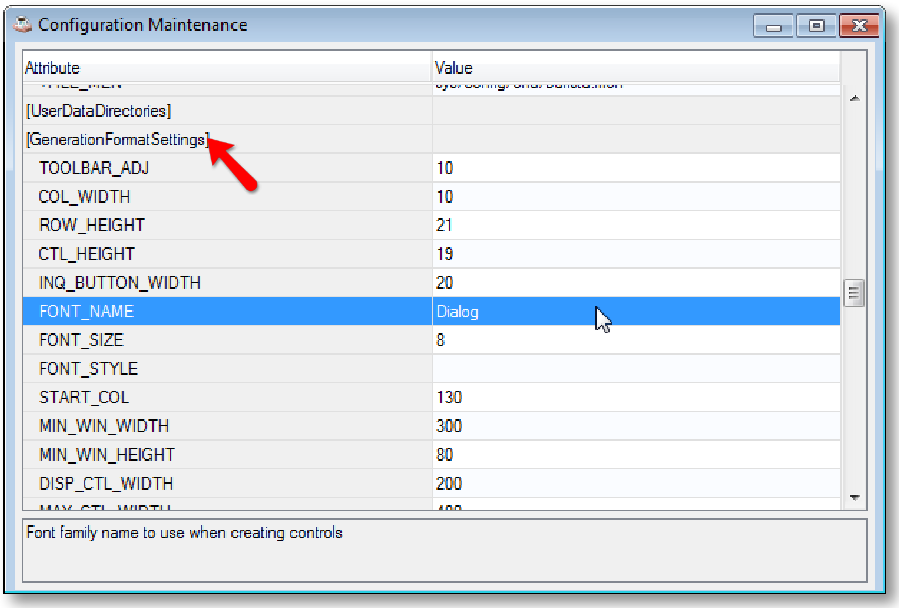
With these readily available resources, you can have a Git repository going and start preserving your precious code today!

Customizing Fonts in Barista

A popular item on the wishlist for the Barista® Application Framework was the request to specify font attributes used on forms built by Barista. While the default font is great for some, other users desired a way to customize the font family and, in particular, the font size in order to improve readability. Barista now allows application developers to specify the font family name, size, and style at three different levels when building forms.

The Default Level

The highest level or "default level" appears under "Barista Development/ Configuration." Barista uses the settings at this level if not overwritten at the form or column level. About half way down through the list of settings, you can see the block "[GenerationFormatSettings]" as shown in **Figure 1** and the three font related settings FONT_NAME, FONT_SIZE, and FONT_STYLE.



Attribute	Value
[UserDataDirectories]	
[GenerationFormatSettings]	
TOOLBAR_ADJ	10
COL_WIDTH	10
ROW_HEIGHT	21
CTL_HEIGHT	19
INQ_BUTTON_WIDTH	20
FONT_NAME	Dialog
FONT_SIZE	8
FONT_STYLE	
START_COL	130
MIN_WIN_WIDTH	300
MIN_WIN_HEIGHT	80
DISP_CTL_WIDTH	200

Font family name to use when creating controls

Figure 1. Configuration dialog

FONT_NAME – The font family name "Dialog" is the default installed with Barista. It is a font in all Java Virtual Machine (JVM) implementations and maps to an actual font contained on the host system. Refer to Font Mnemonic in the online BASIS Help for a list of fonts delivered with the JVM and BBj®.

FONT_SIZE – The font size is the number of points and set to 8 as the default during a Barista installation.

By Ralph Lance
Software Engineer

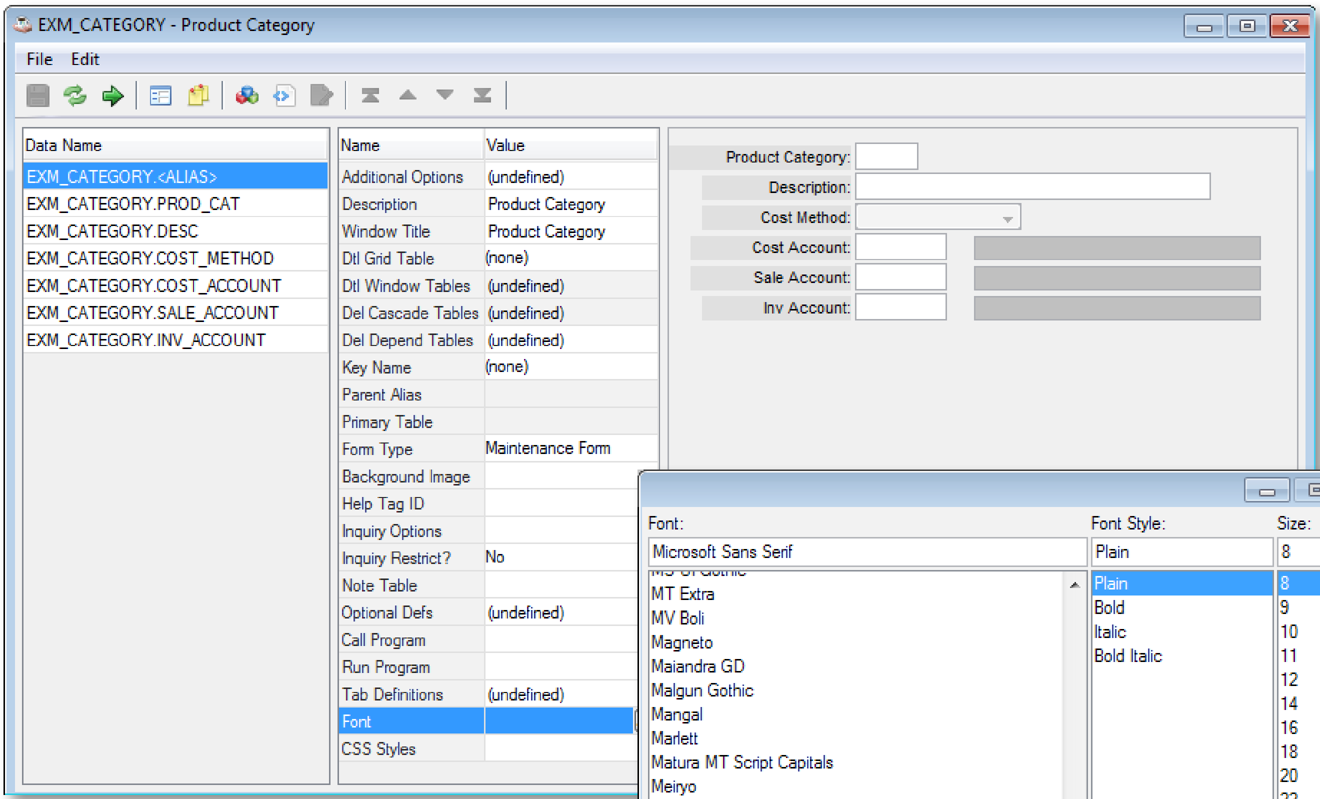
FONT_STYLE – The font style is the numerical value for the style with 0 (PLAIN) being the default. The numerical values may be specified and are additive for combined effects as shown in **Figure 2.**

STYLE	NUMBER
PLAIN	0
BOLD	1
ITALIC	2
UNDERLINE	16

Figure 2. Font values

The Form Level

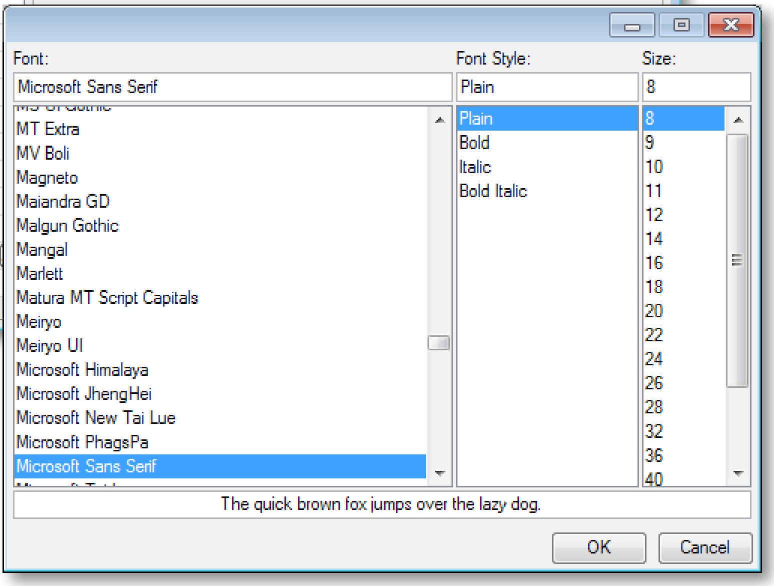
Access the form level via the menu "Barista Development/Form Manager" and double-clicking on a table alias name to load the form definition in the Form Designer. For our demonstration, we'll be using the Barista Examples Product Category alias "EXM_CATEGORY" as seen in **Figure 3**.



Data Name
EXM_CATEGORY.<ALIAS>
EXM_CATEGORY.PROD_CAT
EXM_CATEGORY.DESC
EXM_CATEGORY.COST_METHOD
EXM_CATEGORY.COST_ACCOUNT
EXM_CATEGORY.SALE_ACCOUNT
EXM_CATEGORY.INV_ACCOUNT

Name	Value
Additional Options	(undefined)
Description	Product Category
Window Title	Product Category
Dtl Grid Table	(none)
Dtl Window Tables	(undefined)
Del Cascade Tables	(undefined)
Del Depend Tables	(undefined)
Key Name	(none)
Parent Alias	
Primary Table	
Form Type	Maintenance Form
Background Image	
Help Tag ID	
Inquiry Options	
Inquiry Restrict?	No
Note Table	
Optional Defs	(undefined)
Call Program	
Run Program	
Tab Definitions	(undefined)
Font	
CSS Styles	

Figure 3. Form Designer


Font: Microsoft Sans Serif
Font Style: Plain
Size: 8

MT Extra
MV Boli
Magneto
Maiandra GD
Malgun Gothic
Mangal
Marlett
Matura MT Script Capitals
Meiryo
Meiryo UI
Microsoft Himalaya
Microsoft JhengHei
Microsoft New Tai Lue
Microsoft PhagsPa
Microsoft Sans Serif

Plain
Bold
Italic
Bold Italic

8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 32, 36, 40

The quick brown fox jumps over the lazy dog.

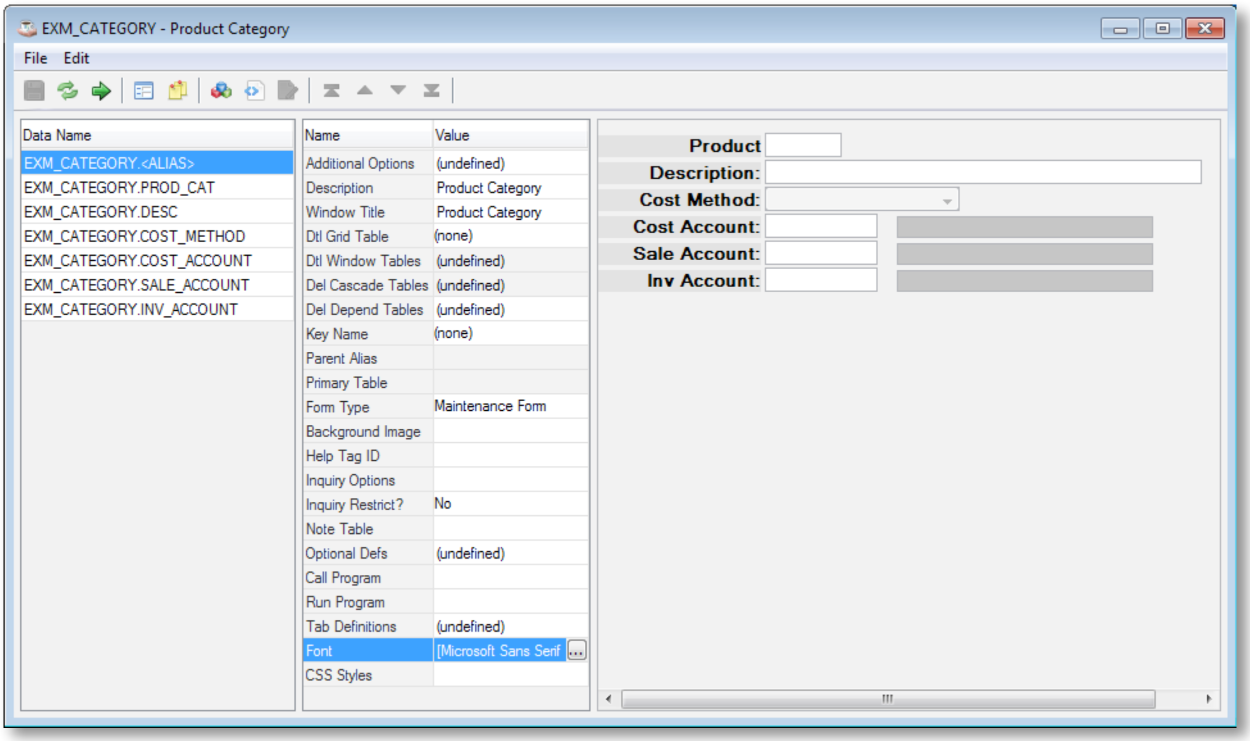
Figure 4. Font Chooser

The first entry in the list of data (column) names on the left is the so-called "Alias" entry. When selecting this item, a list appears of attributes applied to the overall form. Our concentration will be on the attribute named "Font" highlighted in **Figure 3**.

Although one may type in the Font attribute value, there is a more convenient way. By doubling-clicking on the grid cell query button [...] containing the ellipsis to the right of the "Font" cell shown in **Figure 3**, the Font Chooser window appears as shown in **Figure 4**.

The Font Chooser returns the chosen font family, size, and style in a special format that Barista then saves and parses for use by all labels and input controls on the form, assuming that fonts aren't set at the column level. The format of the font data is `[<font name>,<style>,<size>]`, for example `[Microsoft Sans Serif,1,10]`.

After changing the form font attribute, simply refresh the graphical representation of the form to show the effects of the new font as seen in **Figure 5.** Running the form at this point shows that the field labels and values in the input controls reflect the form font.



EXM_CATEGORY - Product Category

File Edit

Data Name
EXM_CATEGORY.<ALIAS>
EXM_CATEGORY.PROD_CAT
EXM_CATEGORY.DESC
EXM_CATEGORY.COST_METHOD
EXM_CATEGORY.COST_ACCOUNT
EXM_CATEGORY.SALE_ACCOUNT
EXM_CATEGORY.INV_ACCOUNT

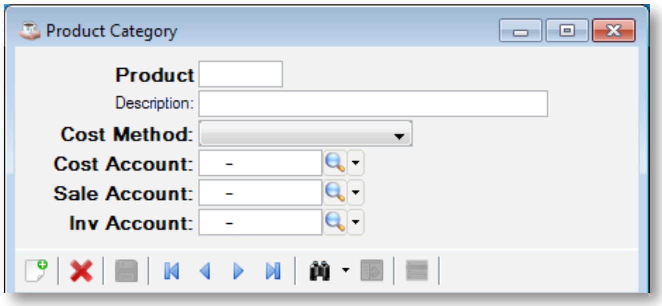
Name	Value
Additional Options	(undefined)
Description	Product Category
Window Title	Product Category
Dtl Grid Table	(none)
Dtl Window Tables	(undefined)
Del Cascade Tables	(undefined)
Del Depend Tables	(undefined)
Key Name	(none)
Parent Alias	
Primary Table	
Form Type	Maintenance Form
Background Image	
Help Tag ID	
Inquiry Options	
Inquiry Restrict?	No
Note Table	
Optional Defs	(undefined)
Call Program	
Run Program	
Tab Definitions	(undefined)
Font	[Microsoft Sans Serif
CSS Styles	

Product
Description:
Cost Method:
Cost Account:
Sale Account:
Inv Account:

Figure 5. Form refreshed after font changes

The Column Level

The most specific level in which to set a font is the field or column level. Similar to the form alias, just select a column either from the list of data names on the left or by clicking the field in the graphical frame, and enter or choose the "Font" attribute for the column. If the form font is set to **`[Microsoft Sans Serif,1,10]`** as **Figure 5** illustrates, then changing the "Description" column to **`[Microsoft Sans Serif,0,8]`** and running the form would result in the change shown in **Figure 6**.



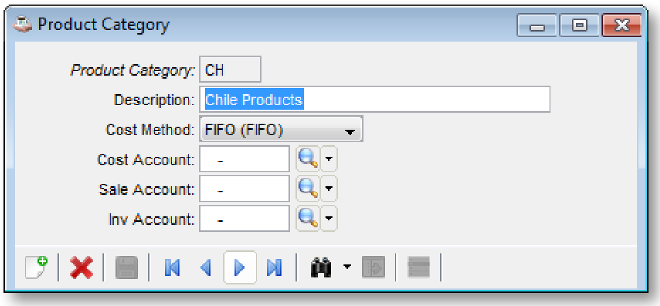
Product Category

Product	
Description:	
Cost Method:	
Cost Account:	-
Sale Account:	-
Inv Account:	-

Figure 6. Form with a single column font change

Removing Font Data

The form and column font attribute values are editable grid cells, so their value is removable by simply double-clicking in the cell, pressing [Delete] and then [Enter]. Don't forget to rebuild the form for any changes to take effect at runtime. Doing so for the form and column fonts we set in this article returns the form back to the original that uses the default font attributes from the Barista configuration (**Figure 7**).



Product Category

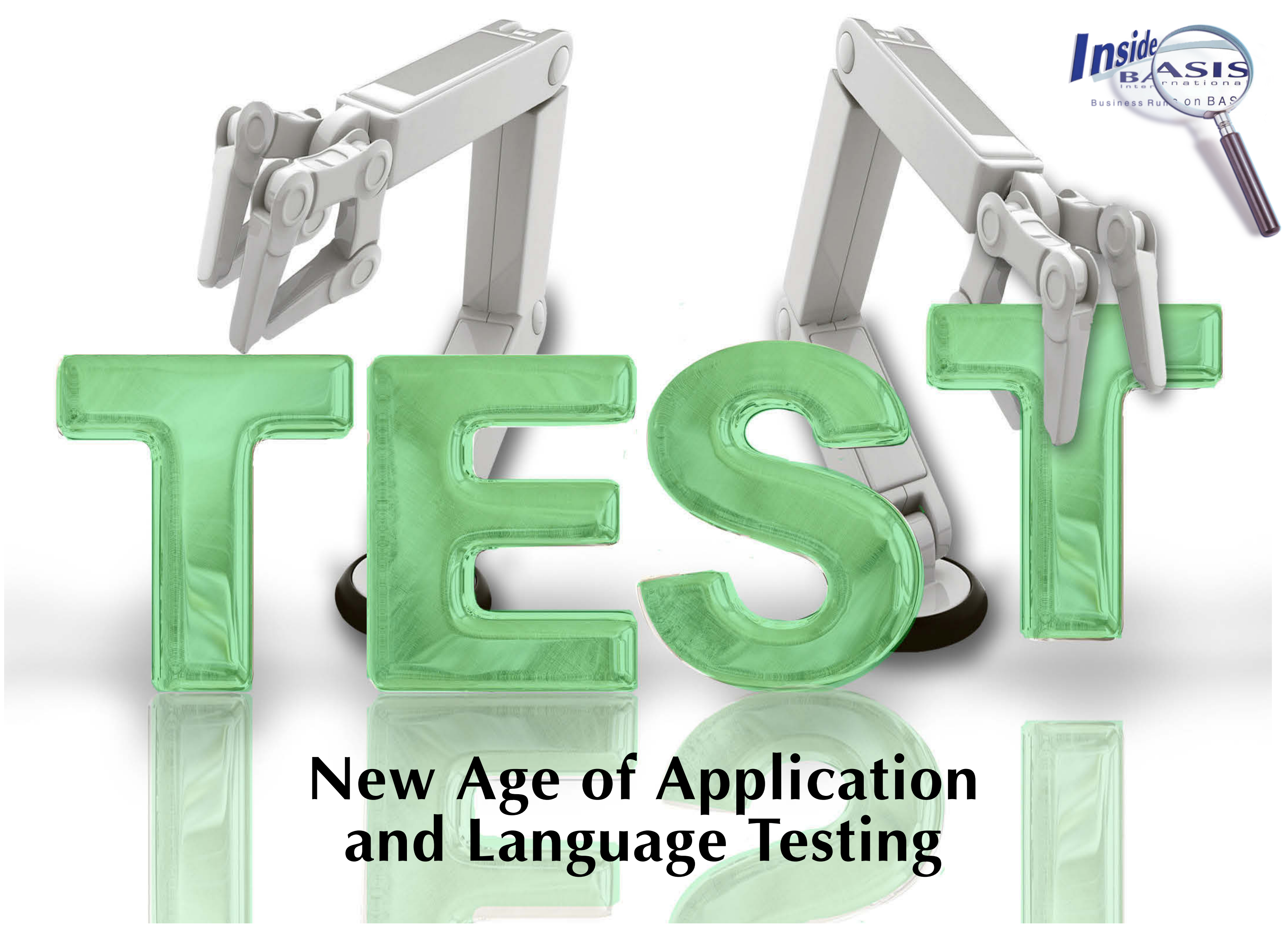
Product Category:	CH
Description:	Chile Products
Cost Method:	FIFO (FIFO)
Cost Account:	-
Sale Account:	-
Inv Account:	-

Figure 7. Original form

Summary

The new capability of setting fonts now gives application developers the power to add character to their Barista forms. Modifying the typeface for an application allows customization of various aspects of the text, including the the font itself, the letter spacing native to the font, the leading or spacing between multiple lines, and so on. By taking advantage of this new Barista feature, your users will be happier and more productive than ever! ■

For more information, go to Font Mnemonic in the online documentation



New Age of Application and Language Testing

The process of creating automated GUI tests in any testing framework usually involves a significant amount of time. To BASIS, this investment of time was well worth the effort as we recently created a new GUI testing strategy for AddonSoftware®. This article looks at different testing processes and specifically, the new paradigm BASIS developed to deliver a higher quality product suite.

Testing...a Look at Both Sides

Manually testing any application from start to finish can take days or weeks depending on the complexity of the application in question. This results in time spent not developing new tests and time not testing new features or functionality. Also, running manual GUI tests does not return feedback in a timely manner to the application developers for them to know how their recent changes affected the overall application. Manual testing is also very prone to easily-missed software issues, human errors, and testing mistakes.

By contrast, automated GUI testing tools such as QFTest can run those same repetitive and laborious GUI tests in a fraction of the time it would take a manual tester to perform the same actions. Automated GUI testing gives the application developers feedback on changes in minutes and hours rather than days and weeks.

Correctly structured and written automated tests are repeatable, dependable, and reliable. By removing the human factor, automated tests remove the mistakes manual testing is prone to make and catch the errors that humans are prone to miss. Automated testing also encourages developers to try new concepts and add additional innovative features within the product. The rapidity and repeatability of automated testing gives reassurance to the software developers that new code is not affecting other vital aspects of the project.

By Robert Del Prete
Quality Assurance Engineer

BASIS...a Look at Their Side Now

To make automated GUI testing possible, BASIS incorporated a number of systems to allow the tests to run nightly in an automated and unsupervised fashion. At the core of the testing model is QFTest, a GUI test tool for Java and the Web. Along with QFTest is an array of other systems – Jenkins, an open-source continuous integration server; the Jenkins Ant plug-in that adds Apache Ant support to Jenkins; and Subversion (SVN), an open source version control system. In addition, BASIS uses the Amazon Elastic Compute Cloud (EC2), a service that provides scalable Windows and UNIX boxes in the cloud. These boxes, or “instances,” install and run

BBj®, AddonSoftware, and the QFTest continually to provide accurate and detailed automated test results of each software build. The BASIS Quality Assurance team then reviews and analyzes these results to help improve the quality of the software and to eliminate bugs and other issues that automated testing uncovers.

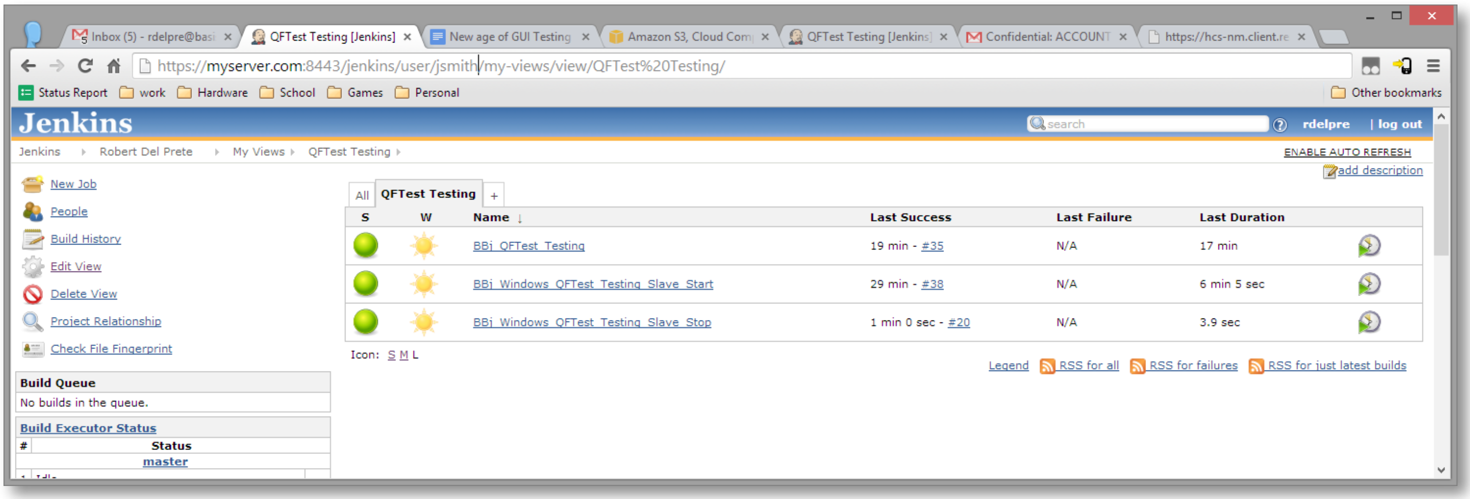
BASIS chose QFTest over other testing solutions for GUI-based testing for several key reasons. QFTest...

- is a cross-platform product,
- is capable of testing in GUI, BUI, and Web Start environments,
- has excellent reporting capabilities,
- offers beginners easy record and playback tests,
- provides access to internal program structures through the use of scripting languages such as Jython and Groovy, and
- integrates easily with Jenkins and Ant.

Integrating Various Technologies

BASIS uses various products and technologies to automate the nightly testing of BBj and AddonSoftware. Using Jenkins, a continuous integration server, we can configure a job that automatically launches a new Amazon EC2 instance using Ant and the Amazon EC2 API Tools. **Figure 1** is an example of the Jenkins dashboard that displays all available tasks that run automated QFTest jobs, the time of each job's "Last Success," "Last Failure," and length of "Last Duration."

In **FIgure 1**, the job named "BBj_Windows_QFTest_Testing_Slave_Start" automatically launches a cloud instance that only costs a few cents per hour to run. Once the stave start job completes, the "BBj_QFTest_Testing" job starts, which checks out the latest BBj/AddonSoftware build from the Amazon's S3 Storage bucket onto the cloud instance it just launched.



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	BBj_QFTest_Testing	19 min - #35	N/A	17 min
●	☀	BBj_Windows_QFTest_Testing_Slave_Start	29 min - #38	N/A	6 min 5 sec
●	☀	BBj_Windows_QFTest_Testing_Slave_Stop	1 min 0 sec - #20	N/A	3.9 sec

Build Queue

No builds in the queue.

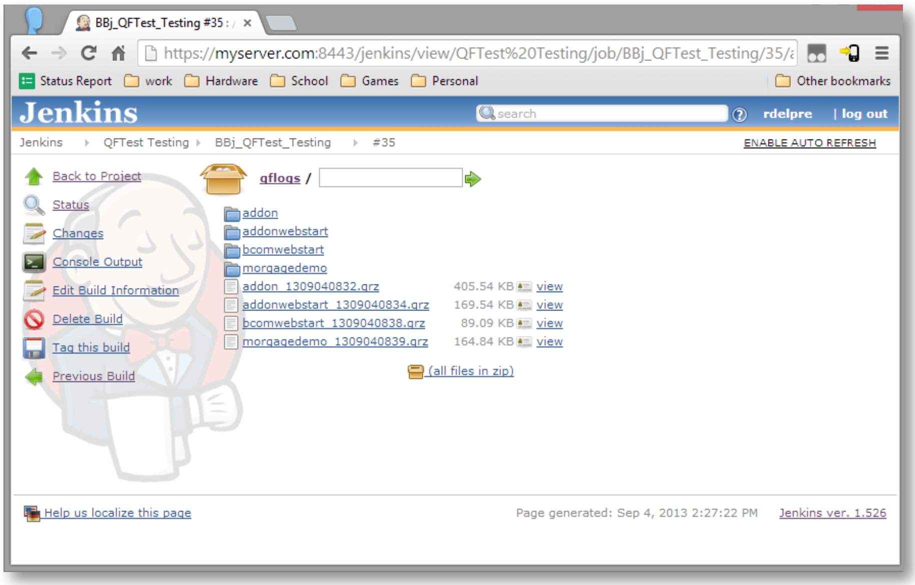
Build Executor Status	
#	Status
	master

Figure 1. Jenkins dashboard QFTest jobs

Next, this job's Ant script installs the BBj/AddonSoftware build and configures it for testing. The Ant script also checks out all relevant QFTest scripts from the BASIS SVN source code control server and delivers them to the cloud machine. Ant then invokes QFTest to run tests in batch mode and delivers the results back to the Jenkins server for review as artifacts (**Figure 2**) of the job. Once the testing is complete, the Jenkins job "BBj_Windows_QFTest_Testing_Slave_Stop" stops the cloud instance.

Analyzing Results

One of the most important aspects of automated testing is conveying accurate and timely results of the test runs during the continuous integration cycle. Jenkins can report the success or failure of the QFTest scripts along with the artifacts from the run.



qflogs /

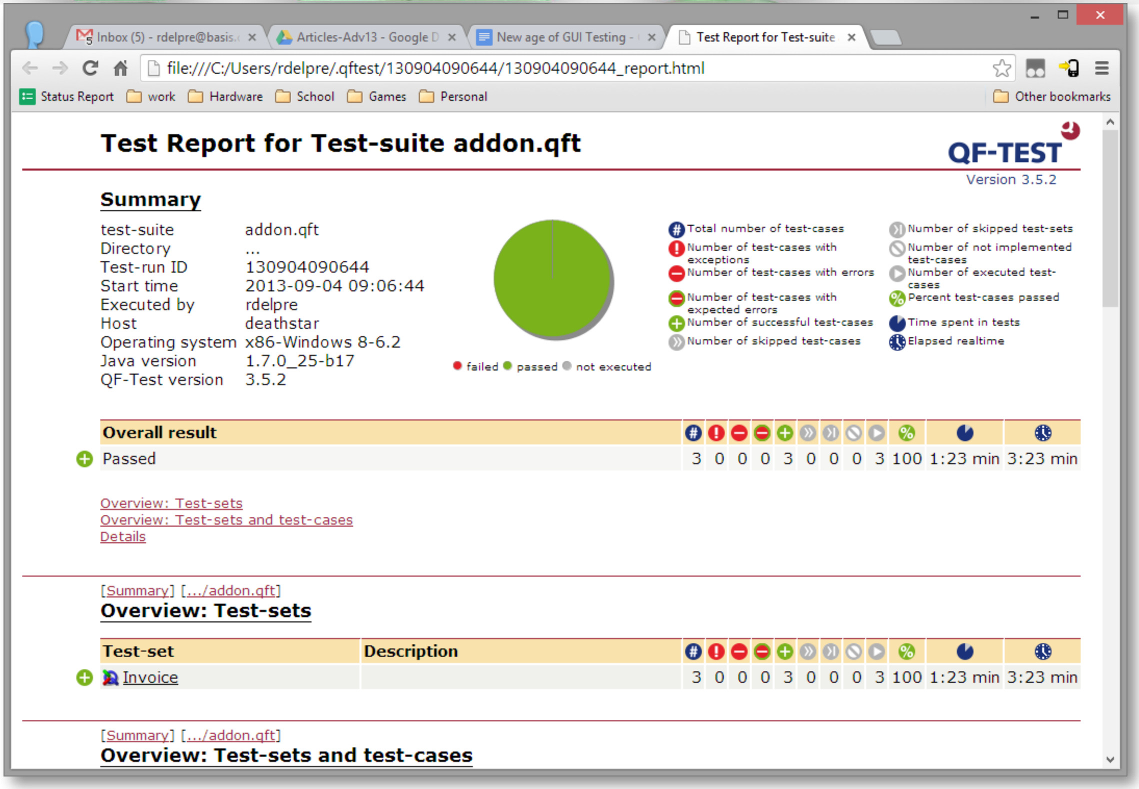
addon
addonwebstart
bcomwebstart
morgagedemo

addon_1309040832.qrz	405.54 KB	view
addonwebstart_1309040834.qrz	169.54 KB	view
bcomwebstart_1309040838.qrz	89.09 KB	view
morgagedemo_1309040839.qrz	164.84 KB	view

(all files in zip)

Page generated: Sep 4, 2013 2:27:22 PM

Figure 2. Jenkins artifacts returned after a successful test run


Inbox (5) - rdelpre@basis. | Articles-Adv13 - Google D | New age of GUI Testing - | Test Report for Test-suite

file:///C:/Users/rdelpre/.qftest/130904090644/130904090644_report.html

Status Report | work | Hardware | School | Games | Personal | Other bookmarks

Test Report for Test-suite addon.qft

QF-TEST
Version 3.5.2

Summary

test-suite	addon.qft
Directory	...
Test-run ID	130904090644
Start time	2013-09-04 09:06:44
Executed by	rdelpre
Host	deathstar
Operating system	x86-Windows 8-6.2
Java version	1.7.0_25-b17
QF-Test version	3.5.2

● failed ● passed ● not executed

# Total number of test-cases	» Number of skipped test-sets
! Number of test-cases with exceptions	⊘ Number of not implemented test-cases
⊖ Number of test-cases with errors	▶ Number of executed test-cases
⊖ Number of test-cases with expected errors	% Percent test-cases passed
⊕ Number of successful test-cases	◔ Time spent in tests
» Number of skipped test-cases	🕓 Elapsed realtime

Overall result	#	!	⊖	⊖	⊕	»	»	⊘	▶	%	◔	🕓
⊕ Passed	3	0	0	0	3	0	0	0	3	100	1:23 min	3:23 min

Overview: Test-sets
Overview: Test-sets and test-cases
Details

[Summary] [.../addon.qft]
Overview: Test-sets

Test-set	Description	#	!	⊖	⊖	⊕	»	»	⊘	▶	%	◔	🕓
⊕ Invoice		3	0	0	0	3	0	0	0	3	100	1:23 min	3:23 min

[Summary] [.../addon.qft]
Overview: Test-sets and test-cases

Figure 3. Sample HTML report of an AddonSoftware test run using QFTest

These artifacts include an .html report (shown in **Figure 3**) and a .qrz file, QFTest's version of a log file that is viewable in the QFTest software.

Together, the .html and .qrz log files provide the information needed to fully analyze all the results of a particular test run. These reports reveal any issues or problems with the test while the QFTest log will help to identify specific failures in the "software under test." The BASIS QA team then sends these problems immediately to software engineers for correction.

Summary

Adding automated functional GUI testing into the continuous integration process provides many advantages. Repeatable and reliable tests provide a method to track improvements in product quality and provide engineers with timely feedback on problems that occur with each change to the product line. Automated GUI testing also helps to ensure a better quality product for BASIS developers and their customers. Specifically, QFTest provides BASIS with a way to include GUI, BUI, and Web Start testing into the build process and integrates well with the Amazon cloud and Jenkins. This truly is a new age of GUI testing at BASIS that will never end because there are always more features to test and new ways to test them! ■

For more information on the cloud, Amazon Elastic Compute Cloud (EC2), etc., read
- *Our Salvation is in the Cloud* at links.basis.com/10basiscloud
- *Perfection in the Cloud* at links.basis.com/11cloud
- *BASIS Survived Amazon Outage* at links.basis.com/12survived

Efficiently S€PArating Your Customers From Their Money

SEPA is an attempt by the European Union to harmonize payment systems in Europe. How will this affect e-commerce and what will BASIS AddonStore customers in Europe have to do now?

Beginning in February 2014 and thanks to SEPA, the "Single Euro Payment Area," all bank-based monetary transactions between market partners in Europe will be processed based on a unified system of account numbers. While the transition was cumbersome for online vendors who were forced to adapt their IT systems, SEPA will create more efficiency and help reduce payment fees in the long run.

For a short period, it looked like SEPA would kill the most popular B-to-C payment option in Germany, the so-called "Lastschrift" or direct debit. The Lastschrift is dealer-friendly, as the dealer can debit the account to which the customer authorized him. SEPA wanted this authorization to be done in writing, per snail mail, which would make the order process clumsy and delay it by a couple of days, not acceptable for customers. But in October 2013, pressure from industrial associations led the SEPA committee to change their plans. Now, as before, it just takes the click of a box to complete the order process.

Direct debit is very popular because it shares the risks equally between the two parties. This arrangement provides for the customers' right to tell their collecting bank to reverse any returned and disputed transactions on their account within a timeframe of six weeks. So, direct debit is easy to handle, risk-free, and involves little technical implementation for online stores.

Payment Options

According to a study by EHI Retail Institute (ehi.org/nc/en), the average online store in Germany offers consumers 6.3 payment procedures to choose from. Among these, the so-called "wallets" are the top of the heap with a presence of 82.4%, the most important provider being PayPal. PayPal lets users link their bank accounts, debit cards, and credit cards directly to one online account. When shopping online, users can then choose to pay directly from their digital account or "digital wallet." This not only means that all of the users' payment information is conveniently available in one place, it also means that the vendor they are buying from never receives their credit card information. PayPal pays the vendor and the PayPal user pays PayPal.

Next in popularity after wallets are the traditional payment methods such as advance payment and cash on delivery (82.3%). Only some 50% of the stores examined offer payment upon invoice, whereas for B-2-B scenarios, "purchase on account" is by far the most common way of payment.

Payment Options Crucial for Customer Confidence

For buyers, the choice of payment options is crucial when it comes to returns. Studies have shown that most German customers believe that there is a connection between the payment option and the willingness and speed by which the retailer will process returns and reimburse them. A majority of consumers believe that PayPal is the safest payment option, followed by credit cards and direct debit, which is favoured by about 10 percent of consumers.

While PayPal seems to be the market leader pretty much everywhere in the industrialized world, online merchants may want to scan their local market because in a mature market such as Germany alone, there are more than 30 providers for wallet services that differ significantly in their market coverage, customer acceptance, and transaction costs. Also, global players Amazon and Google offer their own wallet services with Google Checkout and Checkout by Amazon.

The same goes for credit cards. While payment with your plastic card is readily accepted everywhere in France and Britain, German retailers and customers alike still have their reservations. But if you decide to offer

By Patrick Schnur
European Marketing/PR

AddonStore

AddonStore™ is BASIS' platform for developing e-business solutions, from simple webstores to integrated solutions over several marketing levels. AddonStore stands out because of its stability, flexibility, and performance. It effortlessly manages product lines with several hundred thousand individual products.

AddonStore integrates smoothly with AddonSoftware® by Barista® and is based on open, widespread industry standards - Java, HTML5, CSS, and JavaScript, all independent of platform and operating system choices. As a result of this open architecture, AddonStore is extremely adaptable in areas such as third-party payment solutions, data storage and the design of graphical front ends (stores).

payment by credit card, the choice of which card brands to accept is easy; Mastercard's and Visa's presence is way ahead of their competition.

How to Implement

For the best customer experience, it is necessary to integrate payment options into the order flow of your website as seamlessly as possible. All payment providers offer extensive manuals from which developers can extract the description of the API and information on how to parametrize the interface according to their requirements.

BASIS followed a hassle-free approach for AddonStore customers. Thanks to AddonStore's object-oriented and modularized structure, you don't need to decide at the beginning of your e-commerce project which payment choices you want to offer to your customers. The array of payment systems does not influence your other strategic considerations. PayPal, as the most accepted payment system worldwide, is already implemented by default in the AddonStore. And since the AddonStore is written in BBj®, which is based on the Java Virtual Machine (JVM), pre-written Java code on offer from the respective providers, can be integrated seamlessly into your code.

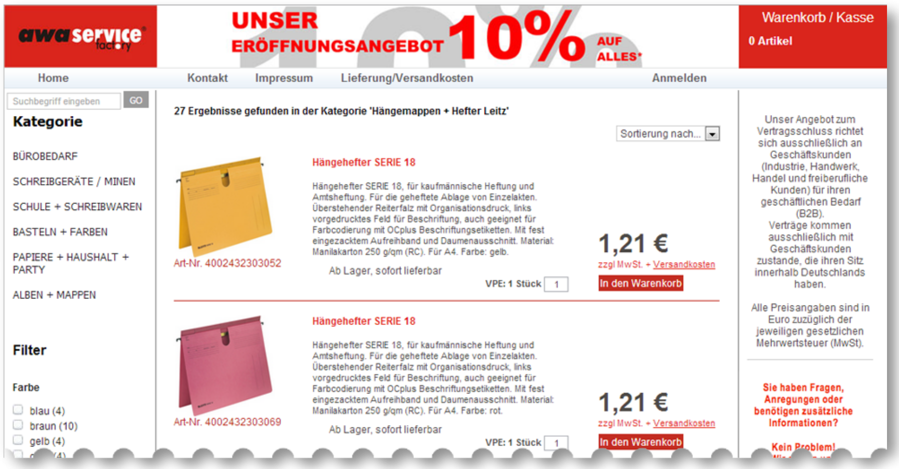
In **Figure 1** two customers who implemented the AddonStore. AWA Service Factory (www.awa-buerobedarf.de) is the sales arm of envelope-maker AWA Couvert in Lower Saxony, Germany. Their AddonStore, appearing in **Figure 1a**, offers their standard range of envelopes as well as stationery and gifts.

Georg Kugelmann (www.georgkugelmann.de), a stationery wholesaler from Hanover, Germany, has been pursuing their e-commerce strategy with the assistance of BASIS Europe since their first online store based on PRO/5® launched in 2005. **Figure 1b** shows the third generation of their Papercompetence store, created after a major relaunch in BBj/BUI.

Market Experience

The AddonStore has been the core of several customer projects BASIS Professional Service has implemented in the U.S. and Europe.

According to Stephan Wald, BASIS Europe's Director of Technical Service, "*All banks, credit card clearing firms, and wallet providers offer ways to integrate their payment services via their APIs. In practice, both our European and U.S. customers saw all their customer's needs covered by integrating Paypal and Paypal online, alongside the direct payment options that don't need much technical implementation in the first place.*"



awaservice factory

UNSER ERÖFFNUNGSANGEBOT 10% AUF ALLES*

Warenkorb / Kasse
0 Artikel

Home | Kontakt | Impressum | Lieferung/Versandkosten | Anmelden

Suchbegriff eingeben GO

Kategorie

BÜROBEDARF
SCHREIBGERÄTE / MINEN
SCHULE + SCHREIBWAREN
BASTELN + FARBEN
PAPIERE + HAUSHALT + PARTY
ALBEN + MAPPEN

Filter

Farbe
blau (4)
braun (10)
gelb (4)

27 Ergebnisse gefunden in der Kategorie 'Hängemappen + Hefter Leitz'

Sortierung nach...

Hängehefter SERIE 18

Hängehefter SERIE 18, für kaufmännische Heftung und Amtsheftung. Für die geheftete Ablage von Einzelakten. Überstehender Reiterfalz mit Organisationsdruck, links vorgedrucktes Feld für Beschriftung, auch geeignet für Farbcodierung mit OCplus Beschriftungsetiketten. Mit fest eingezacktem Aufreihband und Daumenausschnitt. Material: Manilakarton 250 g/qm (RC). Für A4. Farbe: gelb.

Art-Nr. 4002432303052

Ab Lager, sofort lieferbar

VPE: 1 Stück 1

1,21 €
zzgl MwSt. + Versandkosten
In den Warenkorb

Hängehefter SERIE 18

Hängehefter SERIE 18, für kaufmännische Heftung und Amtsheftung. Für die geheftete Ablage von Einzelakten. Überstehender Reiterfalz mit Organisationsdruck, links vorgedrucktes Feld für Beschriftung, auch geeignet für Farbcodierung mit OCplus Beschriftungsetiketten. Mit fest eingezacktem Aufreihband und Daumenausschnitt. Material: Manilakarton 250 g/qm (RC). Für A4. Farbe: rot.

Art-Nr. 4002432303069

Ab Lager, sofort lieferbar

VPE: 1 Stück 1

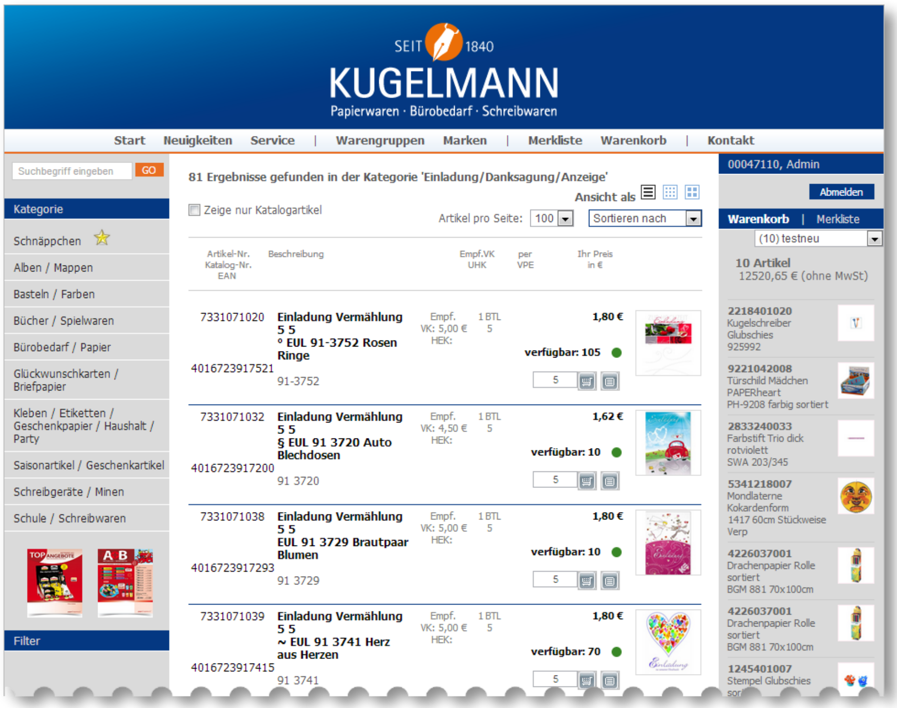
1,21 €
zzgl MwSt. + Versandkosten
In den Warenkorb

Unser Angebot zum Vertragsschluss richtet sich ausschließlich an Geschäftskunden (Industrie, Handwerk, Handel und freiberufliche Kunden) für ihren geschäftlichen Bedarf (B2B).
Verträge kommen ausschließlich mit Geschäftskunden zustande, die ihren Sitz innerhalb Deutschlands haben.

Alle Preisangaben sind in Euro zuzüglich der jeweiligen gesetzlichen Mehrwertsteuer (MwSt).

Sie haben Fragen, Anregungen oder benötigen zusätzliche Informationen?

Kein Problem!



SEIT 1840
KUGELMANN
Papierwaren · Bürobedarf · Schreibwaren

Start | Neuigkeiten | Service | Warengruppen | Marken | Merkliste | Warenkorb | Kontakt

Suchbegriff eingeben GO

Kategorie
Schnäppchen
Alben / Mappen
Basteln / Farben
Bücher / Spielwaren
Bürobedarf / Papier
Glückwunschkarten / Briefpapier
Kleben / Etiketten / Geschenkpapier / Haushalt / Party
Saisonartikel / Geschenkartikel
Schreibgeräte / Minen
Schule / Schreibwaren

Filter

81 Ergebnisse gefunden in der Kategorie 'Einladung/Danksagung/Anzeige'

Ansicht als

Zeige nur Katalogartikel

Artikel pro Seite: 100 | Sortieren nach

Artikel-Nr. Katalog-Nr. EAN	Beschreibung	Empf.VK UHK	per VPE	Ihr Preis in €
7331071020 4016723917521	**Einladung Vermählung 5 5 ° EUL 91-3752 Rosen Ringe** 91-3752	Empf. VK: 5,00 € HEK:	1 BTL 5	**1,80 €** **verfügbar: 105** ● 5
7331071032 4016723917200	**Einladung Vermählung 5 5 § EUL 91 3720 Auto Blechdosen** 91 3720	Empf. VK: 4,50 € HEK:	1 BTL 5	**1,62 €** **verfügbar: 10** ● 5
7331071038 4016723917293	**Einladung Vermählung 5 5 EUL 91 3729 Brautpaar Blumen** 91 3729	Empf. VK: 5,00 € HEK:	1 BTL 5	**1,80 €** **verfügbar: 10** ● 5
7331071039 4016723917415	**Einladung Vermählung 5 5 ~ EUL 91 3741 Herz aus Herzen** 91 3741	Empf. VK: 5,00 € HEK:	1 BTL 5	**1,80 €** **verfügbar: 70** ● 5

00047110, Admin
Abmelden

Warenkorb | Merkliste
(10) testneu

10 Artikel
12520,65 € (ohne MwSt)

2218401020 Kugelschreiber Glubschies 925992
9221042008 Türschild Mädchen PAPERheart PH-9208 farbig sortiert
2833240033 Farbstift Trio dick rotviolett SWA 203/345
5341218007 Mondlaterne Kokardenform 1417 60cm Stückweise Verp
4226037001 Drachenpapier Rolle sortiert BGM 881 70x100cm
4226037001 Drachenpapier Rolle sortiert BGM 881 70x100cm
1245401007 Stempel Glubschies

Figure 1a. The AWA AddonStore (top) **Figure 1b.** The Papercompetence AddonStore build after a relaunch in BBj/BUI (bottom)

At BISCO Industries, a California-based supplier of electronic parts and components with a network of 45 offices throughout the U. S. and Canada, BASIS implemented PayPal Merchant Services so that both a clearing for Paypal account owners, as well as full-scale credit card clearing via Paypal, are possible. German customer paperplace.de, who sells office supplies and stationery to SMCs in a B-to-B scenario, added PayPal to their traditional options of Invoice, Direct Debit, and Cash on Delivery.

Summary

The AddonStore is open to all choices. Providing payment choices can be crucial to the success of your webstore. When it comes to payment options, dealers have a choice of dozens of solutions provided by banks and wallet providers which differ in customer recognition and acceptance. Whatever your choice may be, as a BASIS customer you can be delighted to know that the BASIS webstore module "AddonStore" delivers the most important options by default, and all others are readily available and easily implemented, thanks to the AddonStore's modularized structure. ■

Read about how Bisco Industries implemented AddonStore at links.basis.com/13bisco

BBjToJavadoc Documents Your Masterpiece

How often does documentation get pushed down to the bottom of the list, getting done in a rush after the new or modified code is finished, or perhaps not written completely, accurately, or ever?

Today's BBj® developer can now ***generate*** documentation quickly, easily, and accurately; and formatted in the familiar and modern look and feel of Oracle's Javadoc tool. Like Oracle's Javadoc tool, BASIS' BBjToJavadoc generates API documentation for BBj object-oriented code in HTML format, completely automatically from comments entered within the BBj source code. What a great benefit it is for object-oriented BASIS developers and Java Developers to be able to interact with both Java and BBj documentation presented almost identically!

By Brian Hipple
Quality Assurance Supervisor

BASIS now documents several of our APIs using this new BBjToJavadoc tool, which automatically updates the published documentation as we add, remove, or modify new classes, methods, and fields. For example, documentation for the new Email utility classes now joins BBJasper, Dialog Wizard, and BBjToJavadoc in the Javadoc-like format. In addition to generating more BBj documentation, the BBjToJavadoc utility also includes other very helpful and important enhancements. This article introduces these new BBj 14.0 enhancements that are now available in preview starting with 13.03.

Package Description

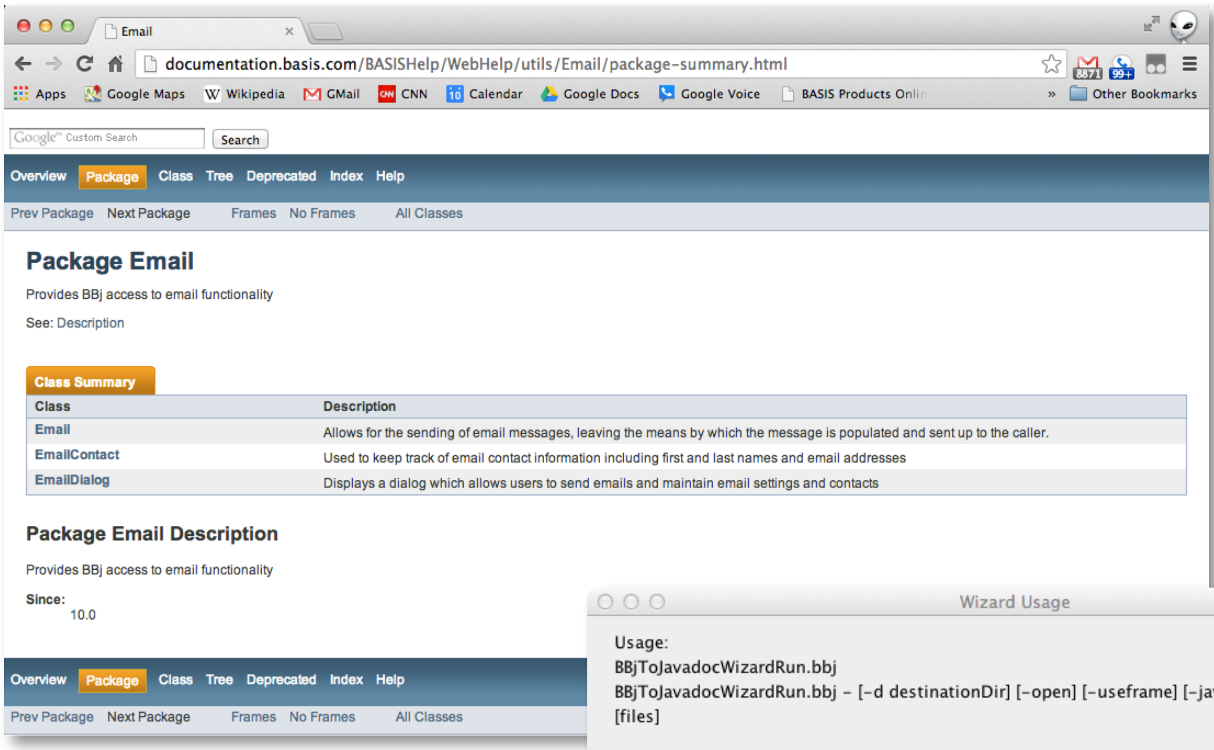
In Java, a "package" is the physical bundling of classes to logical units. In BBj, this traditionally works with the PREFIX and other filesystem-based concepts. When we previously documented BBj code, we set the package to "Unknown." However, we now use the notion of a package in the BBjToJavadoc utility to tag different BBj classes as a logical group to appear together in the documentation. The idea is that one of the source files in the package has package description comment before the package statement, whereas the rest of the modules just have the package statement.

The package statement, which takes the form "REM package [name]", has no programmatic function. However, it is a very powerful documentation statement as a single source directory can support more than one package. The Java paradigm does not allow this, but it fits very well into the BBj paradigm. The associated package Javadoc comment, located before the package statement, supports all Javadoc tags.

Figure 1 shows an example of the package statement and comment in BBj object-oriented source code and **Figure 2** shows the generated documentation.

```
rem /**
rem * Provides BBj access to email functionality
rem * @since 10.0
rem */
rem package Email

rem /** Email
rem * Allows for the sending of email messages, leaving the means by which the message
rem * is populated and sent up to the caller. Attachments are supported.
rem */
class public Email

rem /** EmailContact
rem * Used to keep track of email contact information including first and last names and email addresses
rem */
class public EmailContact

rem /** EmailDialog
rem * Displays a dialog which allows users to send emails and maintain email settings and contacts
rem */
class public EmailDialog
```

Figure 1. Package statement in BBj object-oriented source code with Javadoc comments



documentation.basis.com/BASISHelp/WebHelp/utils/Email/package-summary.html

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

Package Email

Provides BBj access to email functionality

See: Description

Class Summary

Class	Description
Email	Allows for the sending of email messages, leaving the means by which the message is populated and sent up to the caller.
EmailContact	Used to keep track of email contact information including first and last names and email addresses
EmailDialog	Displays a dialog which allows users to send emails and maintain email settings and contacts

Package Email Description

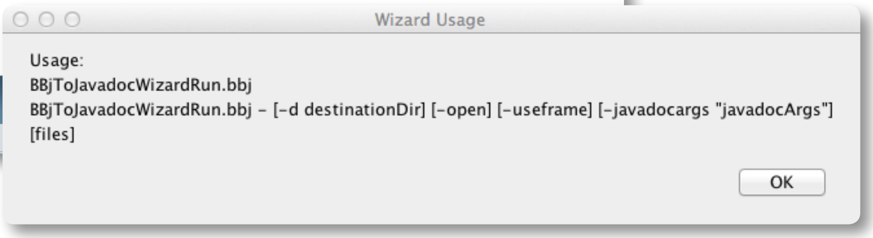
Provides BBj access to email functionality

Since:
10.0

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

Figure 2. Generated Javadoc documentation for the Email package



Wizard Usage

```
Usage:
BBjToJavadocWizardRun.bbj
BBjToJavadocWizardRun.bbj - [-d destinationDir] [-open] [-useframe] [-javadocargs "javadocArgs"]
[files]
```

OK

Figure 3. BBjToJavadoc usage showing new command line arguments

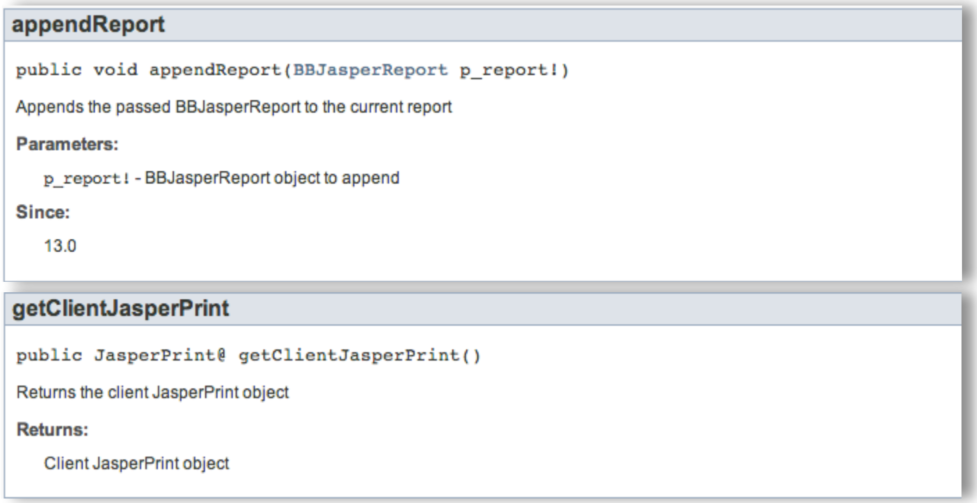
Command Line BBjToJavadoc

One of the most useful enhancements is the ability to run the BBjToJavadoc wizard from the command line. See **Figure 3** for an example of executing the utility with optional command line parameters.

In earlier revisions, this utility could only execute as a GUI wizard. BASIS now incorporates the calling of the command line BBjToJavadoc in its continuous build system to generate the BBj utility documentation. This includes the Email, BBjToJavadoc, DialogWizard, and BBJasper utilities. Anytime a developer checks in a source code change to the SVN archive for one of these utilities, the BASIS build script automatically invokes the BBjToJavadoc utility that generates the updated documentation.

Increased Performance

BASIS engineers devoted much effort to increase the processing speed dramatically in the latest version of the BBjToJavadoc utility. What once took minutes to generate documentation for a large number of source files now measures in mere seconds. In previous versions, the resultant documentation stripped such symbols as the "!" mark and the "@"symbol. BBj now fully supports such concepts as using the ! mark as a suffix for object variable names and the @ symbol in the declaration of client variables. These symbols now show up in the documentation to allow for easy distinction of BBj server and client objects as **Figure 4** shows.



appendReport

```
public void appendReport(BBJasperReport p_report!)
```

Appends the passed BBJasperReport to the current report

Parameters:

`p_report!` - BBJasperReport object to append

Since:

13.0

getClientJasperPrint

```
public JasperPrint@ getClientJasperPrint()
```

Returns the client JasperPrint object

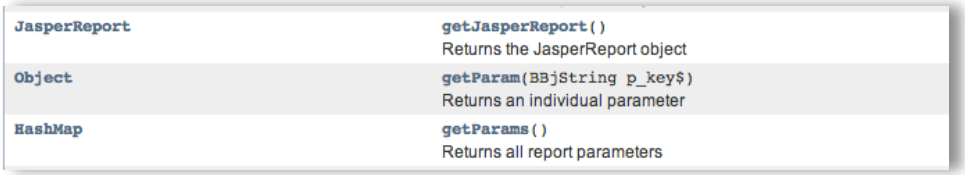
Returns:

Client JasperPrint object

Figure 4. Documentation that includes a BBj server object (top) and client object (bottom)

New Linkage

BASIS now automatically includes linkage to the Java API documentation. When a BBj program uses a Java object as a return value, a parameter to a method, or declares it as a field or variable, BBjToJavadoc creates the links to the associated Java documentation. To establish linkage to a Javadoc set that isn't part of the Java API, add the **`-javadocargs -link`** arguments to instruct the utility to link that documentation to the generated BBj source documentation. BASIS uses this capability in **Figure 5** to link to the Jasper documentation when creating the documentation of the BBJasper utility.



`JasperReport`	`getJasperReport()` Returns the JasperReport object
`Object`	`getParam(BBjString p_key$)` Returns an individual parameter
`HashMap`	`getParams()` Returns all report parameters

Figure 5. Example of the linkage of Java andJasper class documentation

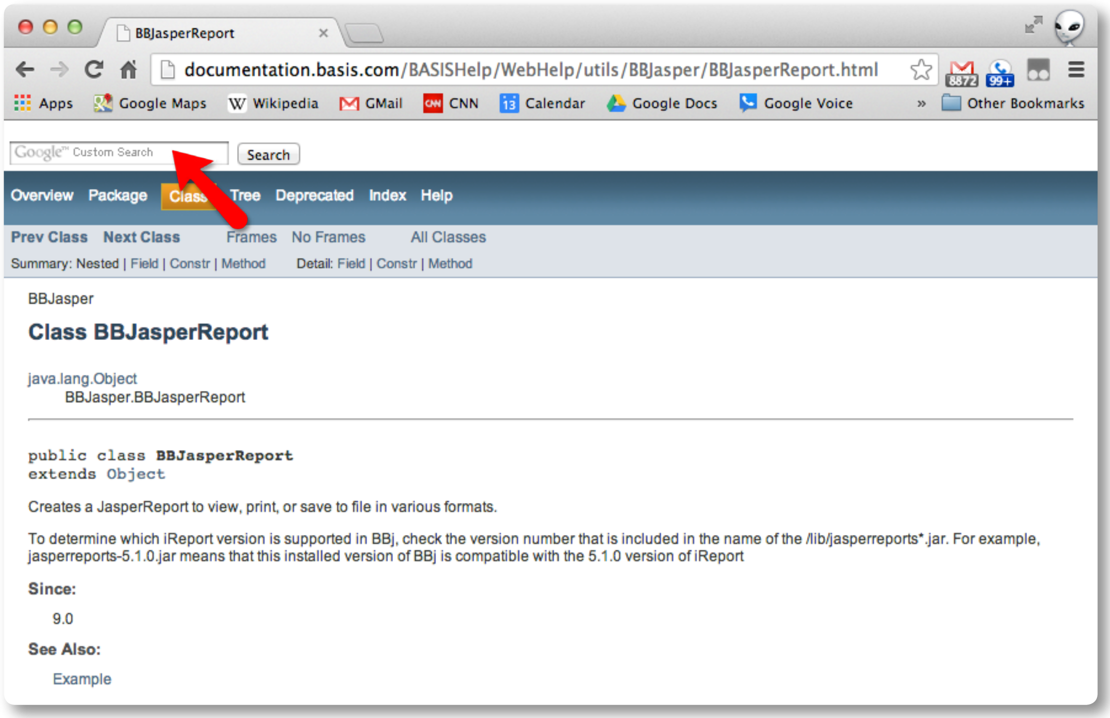
Additionally, developers can add **`@see`** in the source code to embed an example in the file as shown in **Figure 6**.

```
rem /**
rem * Creates a JasperReport to view, print, or save to file in various formats.<p>
rem * To determine which iReport version is supported in BBj, check the version number that is included
rem * in the name of the <bbj install dir>/lib/jasperreports*.jar. For example, jasperreports-5.1.0.jar
rem * means that this installed version of BBj is compatible with the 5.1.0 version of iReport
rem * @since 9.0
rem * @see <a href="BBJasperReportExample.bbj">Example</a>
rem */
class public BBJasperReport
```

Figure 6. Using the **`@see`** tag to include example source code

Google Search

The Javadoc documentation published on the BASIS website now includes our familiar Google search capabilities. This search option appears by simply adding the appropriate header to the documentation, giving the user the ability to type in keywords to search all of the BASIS documentation. **Figure 7** shows the result of a search on "bbjasperreport."



BBJasper

Class BBJasperReport

java.lang.Object
BBJasper.BBJasperReport

```
public class BBJasperReport
extends Object
```

Creates a JasperReport to view, print, or save to file in various formats.

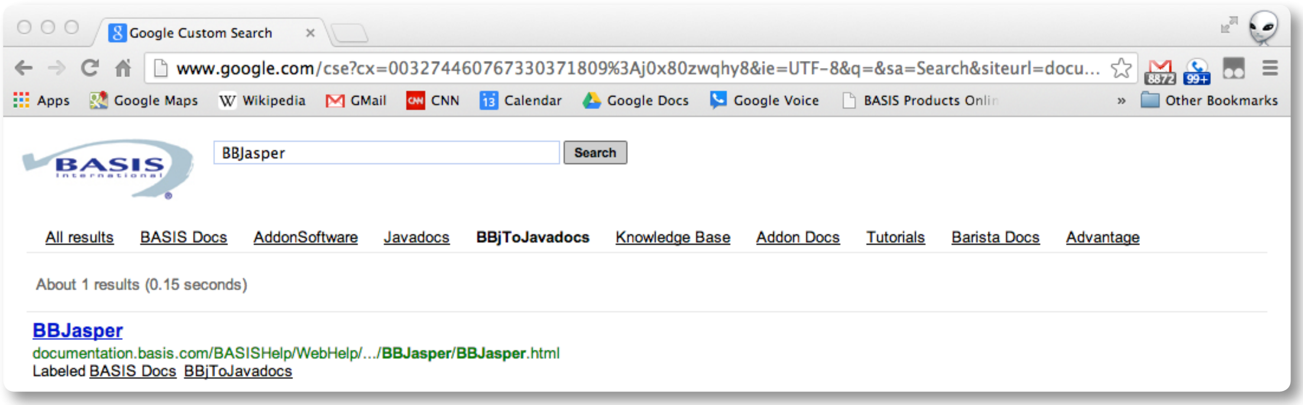
To determine which iReport version is supported in BBj, check the version number that is included in the name of the /lib/jasperreports*.jar. For example, jasperreports-5.1.0.jar means that this installed version of BBj is compatible with the 5.1.0 version of iReport

Since:
9.0

See Also:
Example

Figure 7. The BBJasperReport documentation that displays as a result of a Google Custom Search

BASIS also provides a filter to specify the BBjToJavadoc documentation from the rest of the search filters. **Figure 8** shows the result of this selection.



About 1 results (0.15 seconds)

BBJasper
documentation.basis.com/BASISHelp/WebHelp/.../BBJasper/BBJasper.html
Labeled BASIS Docs BBjToJavadocs

Figure 8. Filter for BBjToJavadoc documentation

Summary

The BBjToJavadoc utility enhancements are great examples of how BASIS goes the extra mile to deliver a more and more useful and efficient tool. This utility allows developers to add documentation as they write their BBj object-oriented code with the code fresh in their minds, resulting in much more accurate and complete documentation. After all, this is far better than writing it after the fact, or even worse, never doing it at all. As most will tell you, it is very hard to maintain or add functionality to poorly documented code. Now, you can *generate* your own documentation, easily and quickly, directly from the code and without the need to format or stylize it, or learn another tool. Your product is now more complete – documented accurately and easily synced with the source code. ■ links.basis.com/**13code**

For more information, read *BBj Documentation is as Easy as JavaDocs* at links.basis.com/12docs

Double Your Pleasure With Eclipse Plug-in Documentation

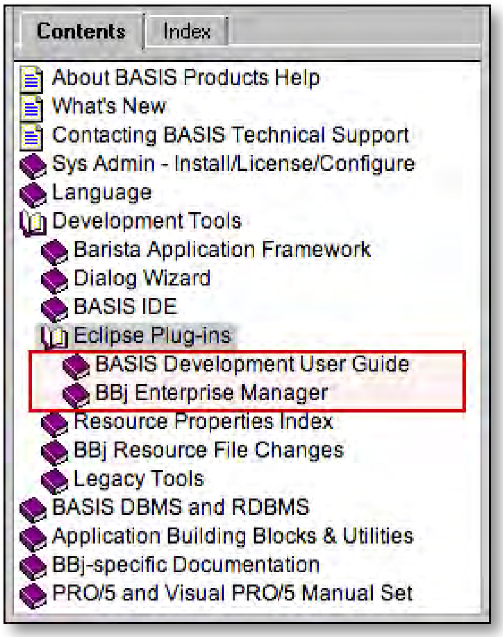
Perhaps you can figure out your new power tool without looking at the manual, but navigating the breadth of features in the new BASIS Eclipse plug-ins will likely be much more challenging. Documenting the new BBj Enterprise Manager and BASIS Development Tools is a hefty project but one that is well underway to help you make the most of these great new advances.

But where are the docs located for easy viewing? If they are inside the plug-in, how do I read about installing it if I don't have it installed?

In BASISHelp

Whether you spend time in the documentation that you downloaded to your local drive or view it online at links.basis.com/basishelp, you will find the Eclipse documentation in the table of contents under *Development Tools* > *Eclipse Plug-ins* as shown in **Figure 1**.

To view the documentation for the BDT and Enterprise Manager plug-in documentation pages directly, go to these easy-to-remember short links: links.basis.com/bdthelp and links.basis.com/emhelp. Each link opens the first topic of the guide



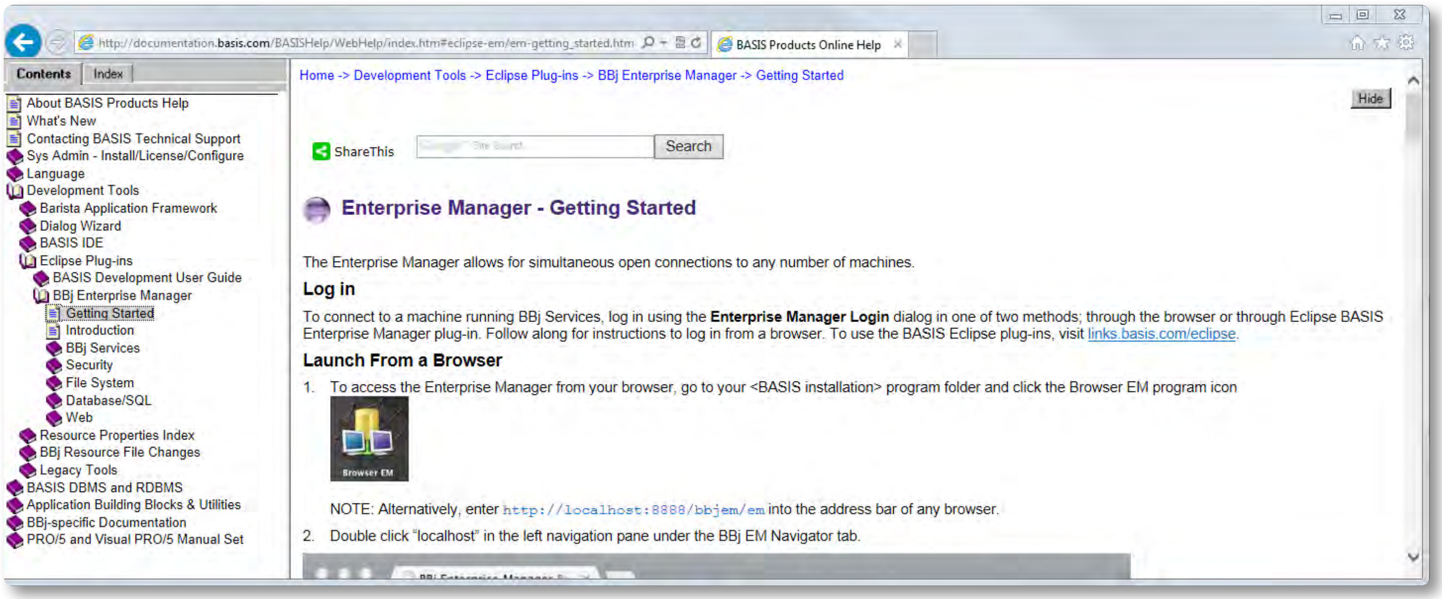
About BASIS Products Help
What's New
Contacting BASIS Technical Support
Sys Admin - Install/License/Configure
Language
Development Tools
- Barista Application Framework
- Dialog Wizard
- BASIS IDE
- Eclipse Plug-ins
  - BASIS Development User Guide
  - BBj Enterprise Manager
- Resource Properties Index
- BBj Resource File Changes
- Legacy Tools

BASIS DBMS and RDBMS
Application Building Blocks & Utilities
BBj-specific Documentation
PRO/5 and Visual PRO/5 Manual Set

Figure 1. Eclipse plug-in docs in BASISHelp

By Susan Darling
Technical/Marketing Writer

inside the BASIS documentation and displays the expanded table of contents entry in the left navigation pane as shown in **Figure 2**. Note the [Hide] button in the upper right corner of the topic; click it to “hide” or “show” the table of contents.


http://documentation.basis.com/BASISHelp/WebHelp/index.htm#eclipse-em/em-getting_started.htm
BASIS Products Online Help
Contents
Index
About BASIS Products Help
What's New
Contacting BASIS Technical Support
Sys Admin - Install/License/Configure
Language
Development Tools
Barista Application Framework
Dialog Wizard
BASIS IDE
Eclipse Plug-ins
BASIS Development User Guide
BBj Enterprise Manager
Getting Started
Introduction
BBj Services
Security
File System
Database/SQL
Web
Resource Properties Index
BBj Resource File Changes
Legacy Tools
BASIS DBMS and RDBMS
Application Building Blocks & Utilities
BBj-specific Documentation
PRO/5 and Visual PRO/5 Manual Set
Home -> Development Tools -> Eclipse Plug-ins -> BBj Enterprise Manager -> Getting Started
Hide
ShareThis
Search
Browser EM

Enterprise Manager - Getting Started

The Enterprise Manager allows for simultaneous open connections to any number of machines.

Log in

To connect to a machine running BBj Services, log in using the **Enterprise Manager Login** dialog in one of two methods; through the browser or through Eclipse BASIS Enterprise Manager plug-in. Follow along for instructions to log in from a browser. To use the BASIS Eclipse plug-ins, visit links.basis.com/eclipse.

Launch From a Browser

1. To access the Enterprise Manager from your browser, go to your <BASIS installation> program folder and click the Browser EM program icon

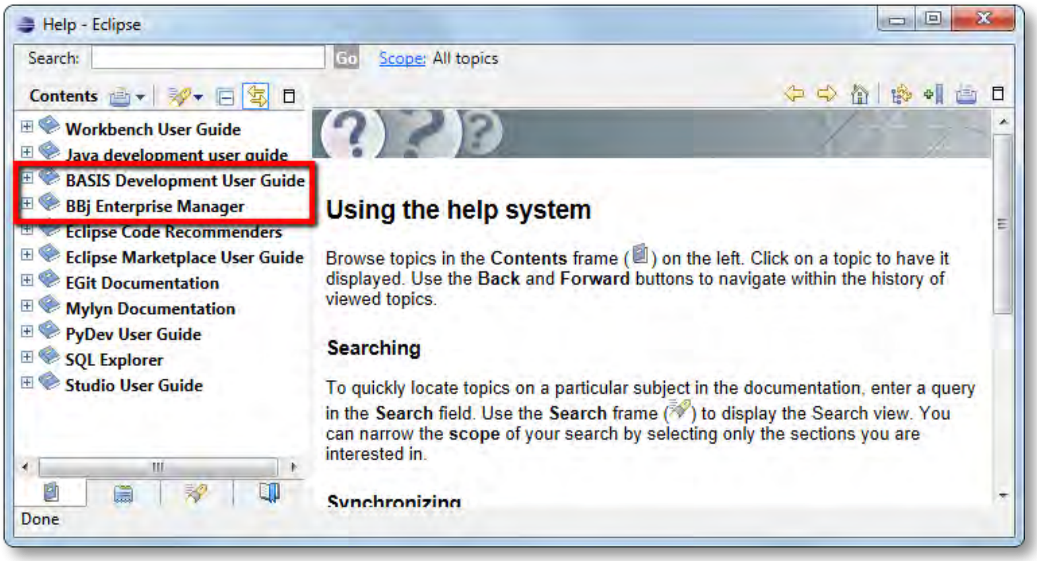
   NOTE: Alternatively, enter `http://localhost:8888/bbjem/em` into the address bar of any browser.

2. Double click “localhost” in the left navigation pane under the BBj EM Navigator tab.

Figure 2. The result of the “emhelp” short link that directly displays the first topic

In Eclipse Help

In addition, all the plug-in topics appear within their respective Eclipse installations. To access the documentation inside Eclipse, click on Help > Help Contents and a window opens that lists the documentation for all of the installed



Using the help system

Browse topics in the **Contents** frame ( ) on the left. Click on a topic to have it displayed. Use the **Back** and **Forward** buttons to navigate within the history of viewed topics.

Searching

To quickly locate topics on a particular subject in the documentation, enter a query in the **Search** field. Use the **Search** frame ( ) to display the Search view. You can narrow the **scope** of your search by selecting only the sections you are interested in.

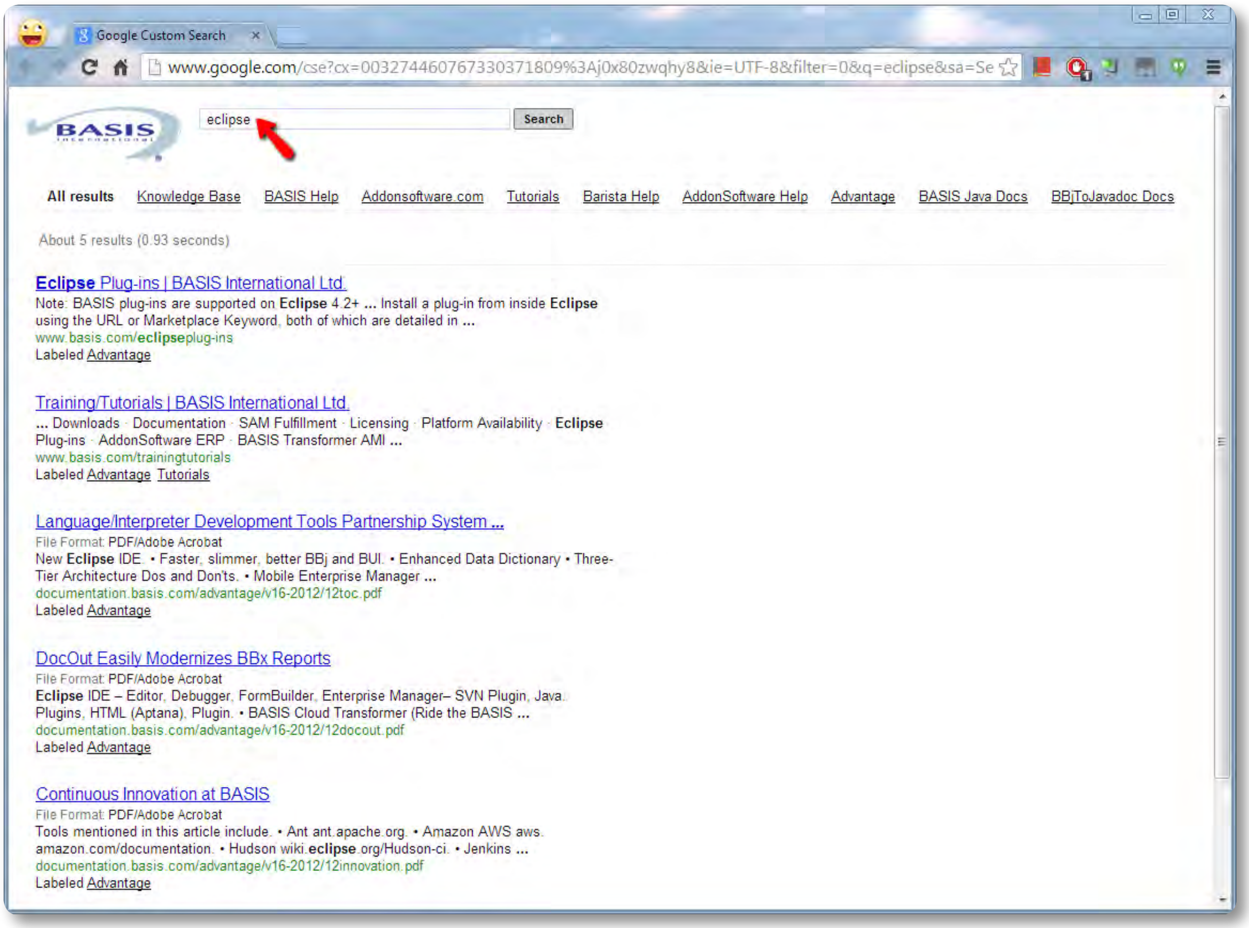
Synchronizing

Figure 3. Documentation for the BASIS plug-ins appearing in the Eclipse Help window

plug-ins. **Figure 3** is an example of this Help window showing the new BASIS plug-ins.

Google Custom Search Engine Help

Lastly, if you don’t know where to look for the help you need for your plug-in question, just ask Google via the Custom Search Engine box on every page of www.basis.com and on every page of the documentation as well. This allows the Google search engine to sort through all of the documentation on the website in tutorials, *The BASIS Advantage* articles, and knowledge base articles, as well as the RoboHelp documentation, to find the best help it can find for your ad hoc question about the Eclipse plug-ins. **Figure 4** shows the search results on the word "eclipse."



About 5 results (0.93 seconds)

Eclipse Plug-ins | BASIS International Ltd.
Note: BASIS plug-ins are supported on **Eclipse** 4.2+ ... Install a plug-in from inside **Eclipse** using the URL or Marketplace Keyword, both of which are detailed in ...
www.basis.com/**eclipse**plug-ins
Labeled Advantage

Training/Tutorials | BASIS International Ltd.
... Downloads · Documentation · SAM Fulfillment · Licensing · Platform Availability · **Eclipse** Plug-ins · AddonSoftware ERP · BASIS Transformer AMI ...
www.basis.com/trainingtutorials
Labeled Advantage Tutorials

Language/Interpreter Development Tools Partnership System ...
File Format: PDF/Adobe Acrobat
New **Eclipse** IDE. • Faster, slimmer, better BBj and BUI. • Enhanced Data Dictionary • Three-Tier Architecture Dos and Don'ts. • Mobile Enterprise Manager ...
documentation.basis.com/advantage/v16-2012/12toc.pdf
Labeled Advantage

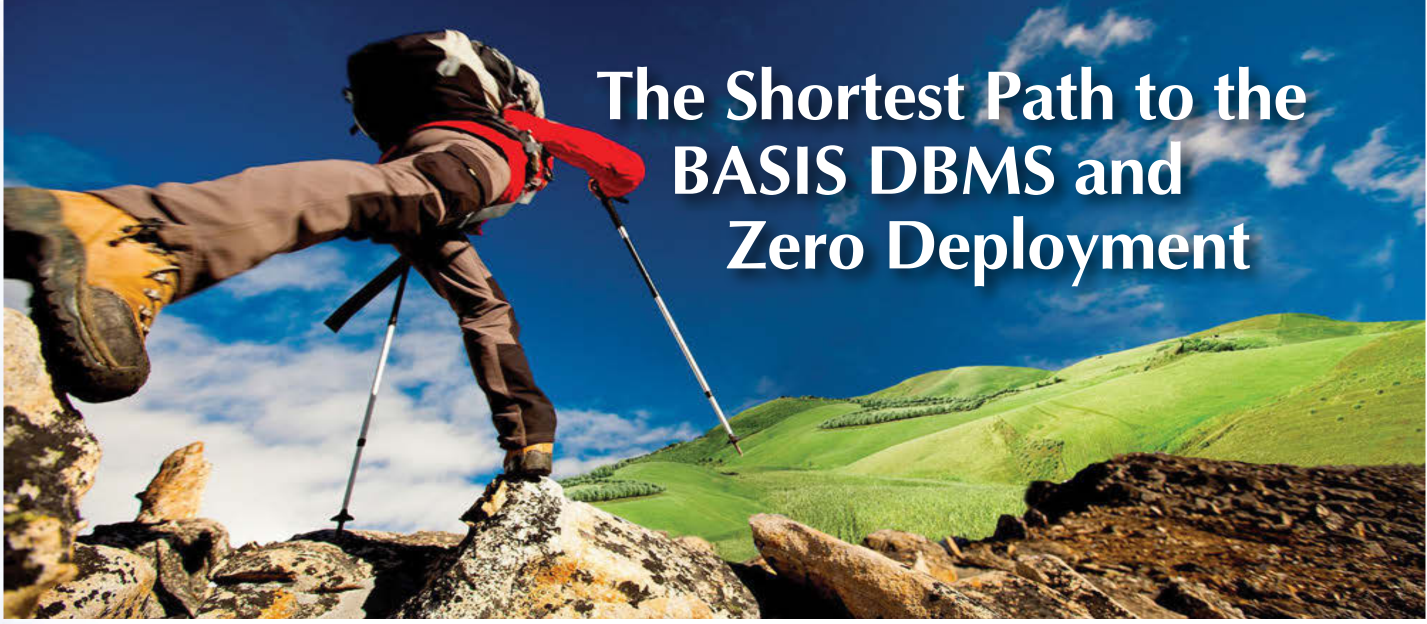
DocOut Easily Modernizes BBx Reports
File Format: PDF/Adobe Acrobat
Eclipse IDE – Editor, Debugger, FormBuilder, Enterprise Manager– SVN Plugin, Java Plugins, HTML (Aptana), Plugin. • BASIS Cloud Transformer (Ride the BASIS ...
documentation.basis.com/advantage/v16-2012/12docout.pdf
Labeled Advantage

Continuous Innovation at BASIS
File Format: PDF/Adobe Acrobat
Tools mentioned in this article include. • Ant ant.apache.org. • Amazon AWS aws.amazon.com/documentation. • Hudson wiki.**eclipse**.org/Hudson-ci. • Jenkins ...
documentation.basis.com/advantage/v16-2012/12innovation.pdf
Labeled Advantage

Figure 4. Searching on "eclipse" from basis.com

Summary

So whichever is your pleasure, you easily have double the access to the documentation for the new BASIS Eclipse plug-ins; in the BASIS Help and within Eclipse. As we continue to complete and expand the various topics, both formats are updated and in sync with the latest text. The new BASIS tools in the Eclipse plug-ins are extremely powerful and very sophisticated so don't be caught wondering how they work...depend on their "manuals" to put these tools to work for you and reap the benefits they offer in your day-to-day development. Double your pleasure! ■



The Shortest Path to the BASIS DBMS and Zero Deployment

In my daily interactions with customers, I see deployments in desperate need of modernization. A monolithic PRO/5® deployment limps along as users are added to an aging UNIX server. A Visual PRO/5®-mapped drive deployment struggles with poor performance and the convolutions of Universal Access Control as the administrator adds new Windows 8 clients. Knowing how easily BBj® eliminates such problems, I have to ask, "Have you considered moving to BBj"? BBj makes it possible to deploy thin clients to desktops without manually installing a terminal emulator or interpreter. With BBj, deploying a thin client is as easy as clicking on a URL that arrived via email. BBj also makes it possible to expand from a simple single-server deployment to a heterogeneous set of multiple servers, distributing the load and providing redundancy.

Triggers, stored procedures, scheduled tasks, data replication, data change auditing, etc. are many of the excellent reasons to upgrade to BBj but, for some, the idea of making the transition from Visual PRO/5 or PRO/5 to BBj can seem as daunting as scaling the sheer face of a cliff. The good news is that the imagined cliff is really just a series of gentle slopes consisting of only three logical steps; each of which will bring real benefits to your deployment.

Step 1. Introduce a networking layer into your configuration

The first step to a BBj deployment doesn't involve BBj at all: Access all data via the PRO/5 Data Server®. While the PRO/5 Data Server is a common element in most Visual PRO/5 deployments, it's not one we often see in typical monolithic PRO/5 deployments.

Adding a data server to your deployment introduces the networking layer into your configuration, which is the key to the flexibility and scalability of enterprise-level deployments. Now it's possible to turn that monolithic PRO/5 deployment into a two-tier deployment; distributing the database management portion of the load onto a database management server (DBMS).

Enable data server access on the PRO/5 or Visual PRO/5 interpreter by setting setopts byte 4 bit $20$:

```
setopts $00000020$
```

The code syntax for opening files directly via the data server is:

```
open(1)"/<server,port=1100>/path/to/filename"
```

Or you could allow the prefix in the config.bbx to locate the file on the database machine:

```
PREFIX </server,port=1100>/path/to/
open(1)"filename"
```

Port 1100 is the default port for the PRO/5 Data Server. For optimal performance, BASIS recommends one data server prefix for data on the interpreter with canonical paths to the data directories specified in the PRO/5 Data Server prefix. For example:

PRO/5 prefix:

```
</server,port=1100>
```

PRO/5 Data Server prefix:

`/usr/local/path/to/data/dir/1/, /usr/local/path/to/data/dir,2/` *…(etc.)*

This ensures only one connection is made to the PRO/5 Data Server for any given file.

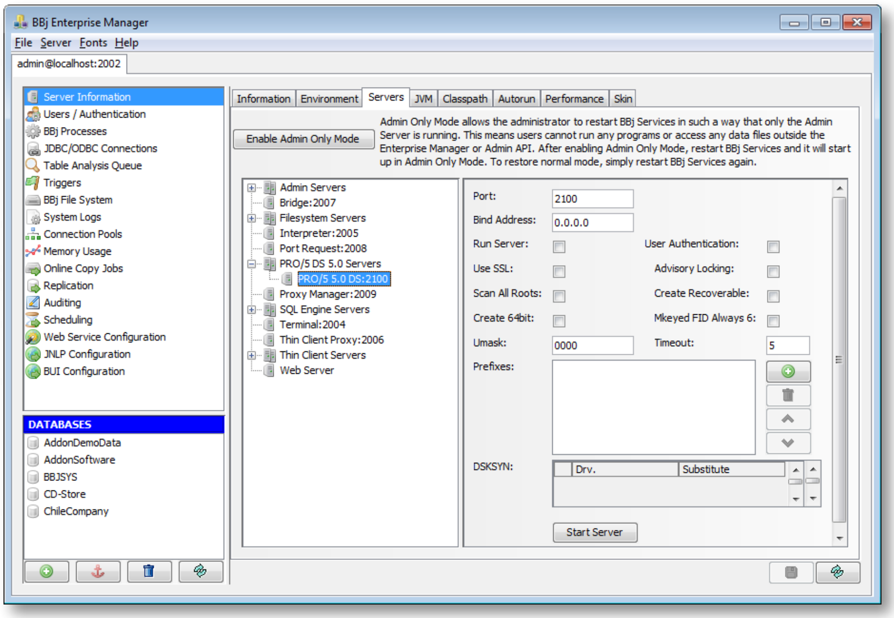
By Bruce Gardner
Technical Support Supervisor

With the introduction of an additional network layer, latency becomes a consideration. While the latency is unlikely to be felt by individual clients, it might be noticeable in batch jobs. This is a good time to run your batch jobs with SETTRACE enabled, evaluate the results in Performance Analyzer, and modify the code to achieve the requisite batch job performance.

Step 2. Switch to the BBj PRO/5 Data Server

If you're looking to retain your PRO/5 or Visual PRO/5 interpreters and just take advantage of all that the BASIS DBMS has to offer over the PRO/5 filesystem, or if you're needing to introduce multi-tier architecture to your application deployment, you will need to introduce a networking layer to your configuration. Once all data is accessed via the PRO/5 Data Server, it's a simple matter of flipping the switch to BBj: Install BBj on the server, enable the BBj PRO/5 Data Server (shown in **Figure 1)**, then change the port in your interpreter prefix to the BBj PRO/5 Data Server port: 2100*.

**You may also leave the prefix port the same and change the BBj PRO/5 Data Server to use port 1100.*



Admin Only Mode allows the administrator to restart BBj Services in such a way that only the Admin Server is running. This means users cannot run any programs or access any data files outside the Enterprise Manager or Admin API. After enabling Admin Only Mode, BBj Services and it will start up in Admin Only Mode. To restore normal mode, simply restart BBj Services again.

Port: 2100
Bind Address: 0.0.0.0
Run Server: ☐ User Authentication: ☐
Use SSL: ☐ Advisory Locking: ☐
Scan All Roots: ☐ Create Recoverable: ☐
Create 64bit: ☐ Mkeyed FID Always 6: ☐
Umask: 0000 Timeout: 5
Prefixes:
DSKSYN:

Figure 1. The BBj PRO/5 Data Server configuration module in Enterprise Manager

Placing control of your data in the hands of BBj enables you to take advantage of several powerful BBj features, including stored procedures, triggers, SQL table analysis, task scheduler, access to additional enhanced BASIS file types, data change auditing, and data replication.

As with the PRO/5 Data Server, the BBj PRO/5 Data Server has server-side prefixing to optimize file OPENs.

Step 3. Switch to BBj

Change the deployment to launch applications with `bbj` instead of `pro5`. With BBj's embedded Jetty Web Server, it is possible to deploy your applications via Java Web Start or the browser user interface (BUI). You don't have to fuss with mapped drive shortcuts when a new Windows 8 client arrives; just send an email with a URL to the user and you're done.

Unlike Visual PRO/5, BBj's interpreter runs on the application server, not the client. This means less back-and-forth processing traffic over the network with only the results presented to the client. The scalability and flexibility of BBj means a deployment can be as small as a single-server or one with many servers. One can swap out or upgrade servers to cloud servers with relative ease. Unlike the PRO/5 Data Server, file I/O via the BBj Filesystem Server scales with additional CPUs, so performance will generally be better in BBj, especially with larger user counts.

Summary

The path from PRO/5 to a modern BBj deployment is not an imposing cliff but a perfectly manageable slope with three logical steps. Each step adds power, flexibility, and scalability to your deployment. Your final BBj destination will bring the ever-growing array of BBj features to your fingertips, allowing you to move your deployment in any direction you like. From the standpoint of the end-users looking in, this translates into a tighter, more simplified user experience. I predict you won't look back! ■

For more information, refer to *Converting to BBj From Earlier Versions of BASIS Products* in the online documentation at links.basis.com/convertingtobbj

You are the missing piece!

Bring your entrepreneurial spirit and add-on your IT know-how to participate in a unique business opportunity!

Find out more at links.addonsoftware.com/opportunity

Meld open source concepts with free-enterprise principles