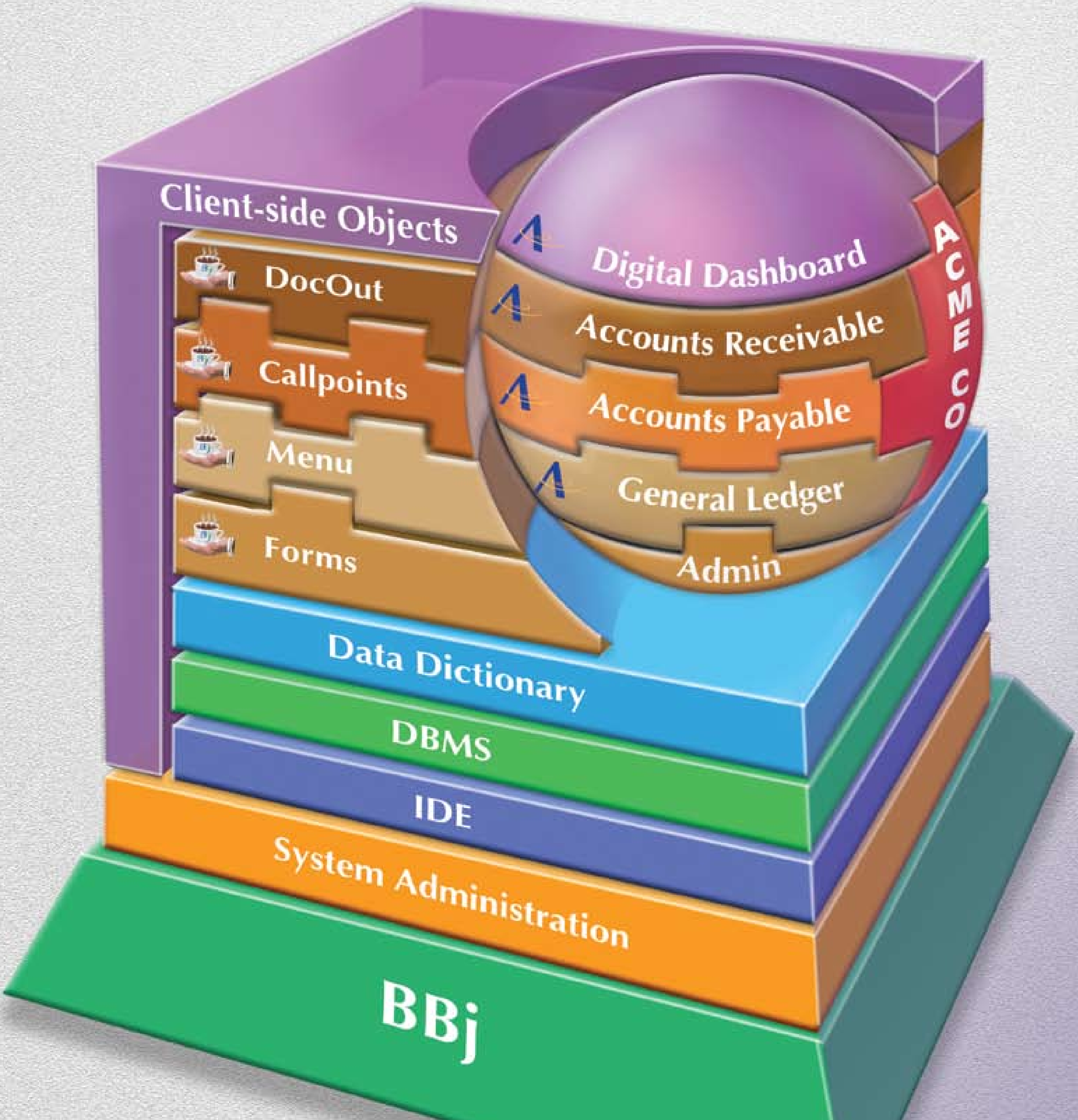
BASIS International Advantage®

Number 1 • Volume 12 • 2008



Document Management Solutions with UnForm®

Production › Delivery › Archiving › Scanning

UnForm is a powerful enterprise document management software solution that seamlessly integrates with any application. The UnForm suite includes laser form and electronic document production, document delivery via email and fax, document archiving and management, and document imaging/scanning. UnForm is a platform independent client server application for Windows®, Unix®, and Linux.

UnForm Laser Forms

UnForm seamlessly integrates with any software application.

- Windows-based graphical design environment
- Eliminate pre-printed forms with laser printer output
- Produce presentation quality reports
- Create e-Documents in Adobe® Acrobat PDF format
- Email e-Documents automatically
- Print bar codes in most symbologies
- Create laser checks with MICR encoding
- Dynamic image conversion and scaling capability
- Database access via ODBC
- Microsoft® Fax Server support
- PCL 5 and Postscript® printer support
- Extensive programmability

Document Archiving and Management

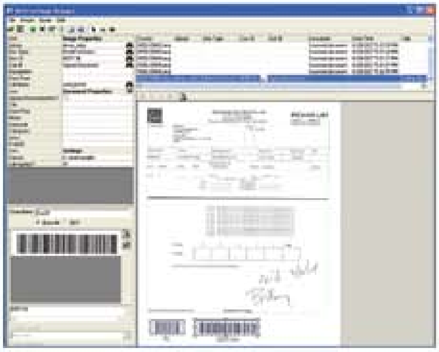
The UnForm Document Archiving and Management component provides the ability to capture, store and retrieve paper-based and electronic documents.

- Rules-based document archiving
- Archive concurrent with document printing
- Store multiple versions of a document
- 10 levels of user defined category indexing
- Document linking control
- Fast web browser-based retrieval
- Client API for application-based retrieval
- Index oriented archive browsing
- Full feature search capability

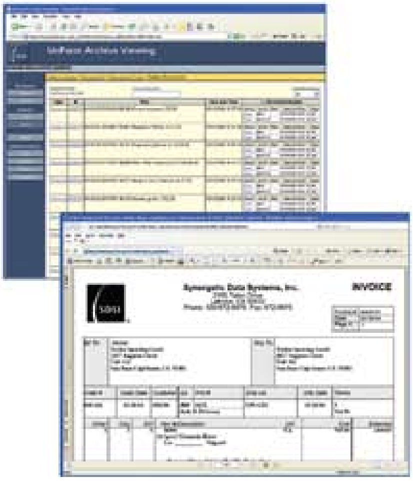
Document Imaging/Scanning

Windows client application that provides a document scanning and importing tool for capture of documents external to the UnForm processing environment.

- Windows client application
- Integrated work environment for image capture and upload
- TWAIN compliant scanning interface
- Multiple property assignment modes
- Barcode and OCR zone detection
- Automatically match or group images with related archive documents
- Extensibility via VB Script



Document Image Manager



Universal web browser document retrieval

Synergetic Data Systems, Inc.
2195 Talon Drive
Latrobe, CA 95682 USA
(800) 446-7374 or (530) 672-9970
Fax: (530) 672-9975
sales@synergetic-data.com
www.synergetic-data.com

WHEN IT COMES TO
DATA PROTECTION,
ASK DATA REFLECTOR™
IS A LIFE SAVER.

6

14

22

28

34

Partnership

Language/Interpreter

DBMS

Development Tools

Applications

System Administration

COLUMNS

🌐 **e-article published exclusively at www.basis.com**

New Leadership, New Tools, New Building Blocks, New Promise

Kevin and I relish the great opportunity and exciting challenge that we assume with our new management positions at BASIS International Ltd. The leadership transition has been a gradual one, and indeed, George Hight continues to contribute to BASIS in a financial administration role while also overseeing our growing operations in Mexico.

One of the things that make our industry truly unique is the sense of wonder and excitement that constant innovation provides. Imagine the consternation and disbelief that you would have generated if you could have stepped back in time 30 years and wandered into the fledgling premises of Microsoft in downtown Albuquerque, armed with a fully-loaded iPhone or a Google Android-powered mobile phone. Of course, you could not demonstrate much of its function; no cell phone network, no Internet, no Wi-Fi, but picture a young Bill Gates' delighted response when you showed him the technology that you carried in your palm. Our latest development framework and run-time engine, Barista®, and its first product, AddonSoftware™ accounting, might have elicited a similar reaction from the engineers in the 1985 BASIS offices in Albuquerque.

As we take up the challenge of leading BASIS to another level, we are excited and delighted to witness the 'rebirth' of AddonSoftware. The promise that Barista and the AddonSoftware building blocks hold for our community is truly amazing. Rapid development, lower costs of maintenance, stronger partnerships, and higher customer and staff satisfaction are just some of the benefits that await our development community when they embrace this new technology.

All of these new products are made possible by their foundational components, the BBj® language, SQL Engine, and BASIS DBMS. This year's new capabilities, like BBj's client-side embedded Java and the enhanced relational database management features of the BASIS DBMS, lay the groundwork for all developers in the BASIS community to add feature and function to their applications. These new features and functions make a significant difference to both large and small applications; from Tesco's 1,900-user system that analyzes huge quantities of data in their Asian retail operations to single user solutions that manage the accounting needs for a small business.

This edition of the *BASIS International Advantage* showcases not only these new features and functions but also delivers practical use cases for recent BASIS technology advances. These and other demonstrations are available for download from the BASIS Web site alongside the current versions of BBj, Barista, and AddonSoftware. Download an evaluation license today; mark the Barista, AddonSoftware and Demo checkboxes, and prepare to be amazed and delighted!

Nico Spence
Chairman & CEO

Dr. Kevin W. King
President & CIO

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

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

Copy Editor *Peggy Lewis*
plewis@basis.com

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

Electronic Production *Amer Child*
achild@basis.com

Printing/Distribution Services
Albuquerque Printing Company

BASIS does not endorse any products mentioned in the *BASIS International Advantage* other than those products licensed by 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/company/trademarks.html. All other product and brand names are trademarks or registered trademarks of their respective companies.

Advantage Subscriptions
www.basis.com/advantage/subs.html

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

General Information
info@basis.com

Freshly Brewed – *Barista Caffeinates Addon Accounting Applications!*

By Chris Hawkins

By now, just about everyone has heard about BASIS' exciting new rapid application development framework – Barista® – as well as the recent release of AddonSoftware™ 8.0, the first major application developed in the Barista Application Framework.

AddonSoftware has a 25-year history as a cost-effective, modular, robust, and fully-integrated enterprise resource planning solution. A strictly character-based application for most of that time, AddonSoftware eventually began the transition to a graphical user interface to take advantage of technological advancements and keep abreast of competitors.

AddonSoftware 7 gave users the opportunity to run in either graphical or character modes. While it provided the more modern look and feel that many users sought, Version 7 was complicated to develop, maintain, support, and especially customize, particularly in terms of the user interface.

With the advent of Barista, AddonSoftware decided to abandon the character mode and "go GUI" only. This was not a trivial undertaking, but considering that it meant remaining in a Business BASIC environment and preserving a large amount of legacy code, it was a desirable option compared to others.

This article reviews the process of creating AddonSoftware 8 and shares some ideas about how to brew a legacy application into an exciting new creation!

Application Analysis

Developing AddonSoftware within the Barista Application Framework offered substantial productivity gains over other alternatives, but the team still had some unique challenges to face. In analyzing the legacy versions, the developers had to determine the best way to take full advantage of the advancements offered by Barista and BBj®, but also preserve tried and true legacy "back-end" code where desirable, and modernize outdated data structures – all while providing an upgrade path for those using the older versions.

The legacy AddonSoftware code fell into roughly five categories: user interface, reports, updates, utilities, and public programs. Barista could handle the most complex of these – the user interface – and would also provide additional functionality in terms of inquiry, drilldowns, and document manipulation. That left the developers' with the manual tasks of porting desired back-end code into a format compatible with Barista and normalizing and converting data files.

At the Front End

Developing the User Interface in Barista

The user interface tends to be the most time-consuming component of any application to develop and maintain. These 'new' GUI user interface design skills often have to be provided by additional hires to the development team or require a substantial divergence of valuable resources from the already expensive resource pool. Using Barista to develop the user interface for AddonSoftware 8 not only enabled developers to create forms faster – several times faster in many cases – but also supplied a consistent look and feel throughout the application without the developers need to acquire a new skill set. In addition, developers were able to take advantage of Barista's inquiry and drilldown capabilities, all without writing so much as a line of code.

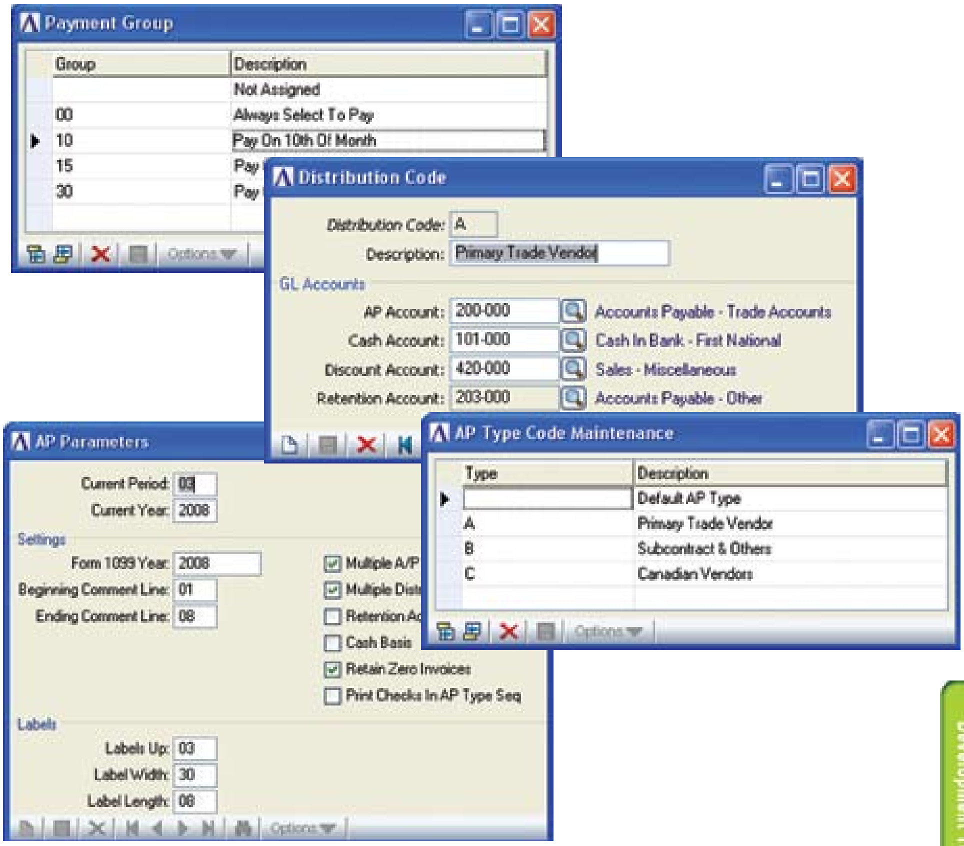
Compared to working in other graphical development environments, Barista's "bottom-up" approach seemed a bit foreign at first. But developers soon discovered the power behind the Barista design. The key to development in this new framework is to define the database elements, and then define tables using those elements. Given a table definition, Barista builds a fully functional form. Really, it can be just that simple, and just that fast.

Chris Hawkins
Software Engineer

continued...

AddonSoftware developers did not have to port any of the legacy user interface programs to Version 8; they developed all of the forms using the Barista framework. As a result, they significantly reduced the amount of code because Barista uses the same handful of programs to run every form. Once the core set of Barista programs is cached, forms run much faster. Also, changes and enhancements to the Barista framework, such as adapting to the constantly evolving GUI standards, are available at once in all forms, and do not have to be propagated manually or by means of a specially written utility.

Forms are easy to alter in the Barista Form Designer. Many of the AddonSoftware forms underwent some amount of cosmetic modification to place fields on tabs, insert group headings, add additional display fields, etc. Most of this work takes just a few minutes to complete. **Figure 1** shows some forms that developers created very quickly, requiring little or no custom code.



Payment Group

Group	Description
	Not Assigned
00	Always Select To Pay
10	Pay On 10th Of Month
15	Pay
30	Pay

Distribution Code

Distribution Code: A
Description: Primary Trade Vendor

GL Accounts

AP Account:	200-000	Accounts Payable - Trade Accounts
Cash Account:	101-000	Cash In Bank - First National
Discount Account:	420-000	Sales - Miscellaneous
Retention Account:	203-000	Accounts Payable - Other

AP Parameters

Current Period: 03
Current Year: 2008

Settings

Form 1099 Year: 2008
Beginning Comment Line: 01
Ending Comment Line: 08

☑ Multiple A/P
☑ Multiple Dist
☐ Retention Ac
☐ Cash Basis
☑ Retain Zero Invoices
☐ Print Checks In AP Type Seq

Labels

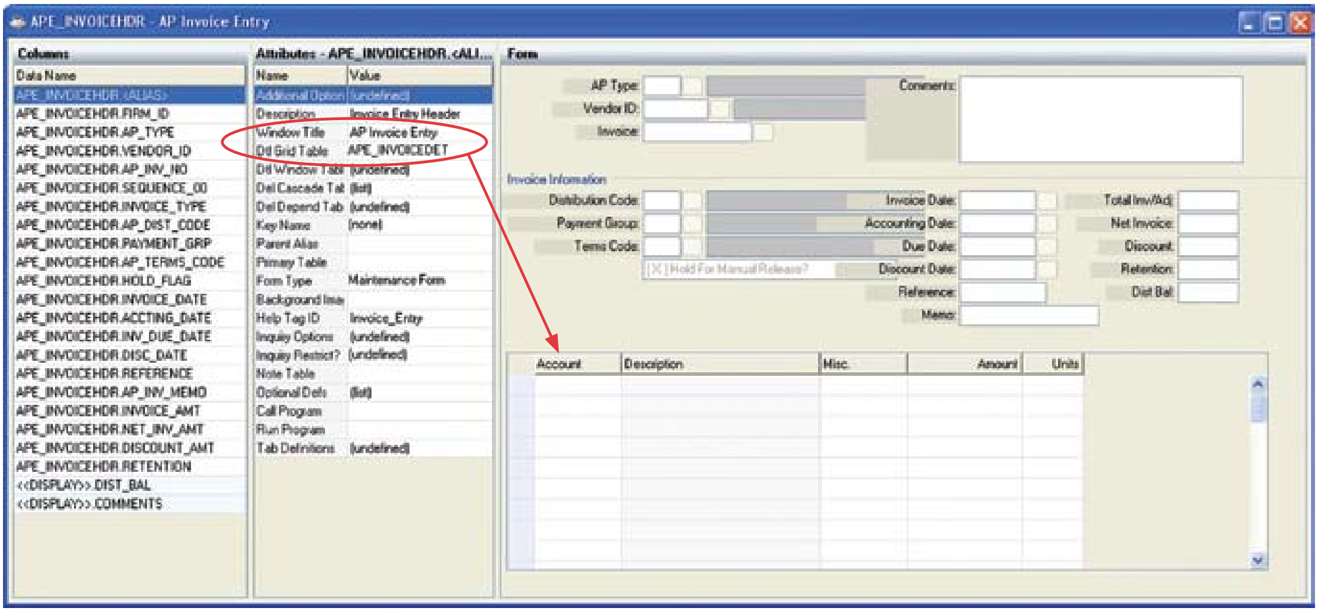
Labels Up: 03
Label Width: 30
Label Length: 08

AP Type Code Maintenance

Type	Description
	Default AP Type
A	Primary Trade Vendor
B	Subcontract & Others
C	Canadian Vendors

Figure 1. Barista enables rapid development of many of an application's less complex forms

Even more powerful is the ability to link forms based on their key structures. AddonSoftware makes use of several forms structured as "header/detail" or "one-to-many." With the header and detail tables already defined, the Form Designer makes it easy to specify that a particular header form contains a detail grid or detail window table. Specifying a detail grid provides a classic one-to-many user interface, with the header information at the top of the form, and the detail information in a grid below. **Figure 2** shows the table definitions and linking mechanism used for Accounts Payable Invoice Entry.



APE_INVOICEHDR - AP Invoice Entry

Columns

Data Name
APE_INVOICEHDR.<ALIAS>
APE_INVOICEHDR.FIRM_ID
APE_INVOICEHDR.AP_TYPE
APE_INVOICEHDR.VENDOR_ID
APE_INVOICEHDR.AP_INV_NO
APE_INVOICEHDR.SEQUENCE_00
APE_INVOICEHDR.INVOICE_TYPE
APE_INVOICEHDR.AP_DIST_CODE
APE_INVOICEHDR.PAYMENT_GRP
APE_INVOICEHDR.AP_TERMS_CODE
APE_INVOICEHDR.HOLD_FLAG
APE_INVOICEHDR.INVOICE_DATE
APE_INVOICEHDR.ACCTING_DATE
APE_INVOICEHDR.INV_DUE_DATE
APE_INVOICEHDR.DISC_DATE
APE_INVOICEHDR.REFERENCE
APE_INVOICEHDR.AP_INV_MEMO
APE_INVOICEHDR.INVOICE_AMT
APE_INVOICEHDR.NET_INV_AMT
APE_INVOICEHDR.DISCOUNT_AMT
APE_INVOICEHDR.RETENTION
<<DISPLAY>>.DIST_BAL
<<DISPLAY>>.COMMENTS

Attributes - APE_INVOICEHDR.<ALI...

Name	Value
Additional Option	(undefined)
Description	Invoice Entry Header
Window Title	AP Invoice Entry
Dtl Grid Table	APE_INVOICEDET
Dtl Window Tabl	(undefined)
Del Cascade Tal	(list)
Del Depend Tab	(undefined)
Key Name	(none)
Parent Alias	
Primary Table	
Form Type	Maintenance Form
Background Ima	
Help Tag ID	Invoice_Entry
Inquiry Options	(undefined)
Inquiry Restrict?	(undefined)
Note Table	
Optional Defs	(list)
Call Program	
Run Program	
Tab Definitions	(undefined)

Form

AP Type:
Vendor ID:
Invoice:
Comments:

Invoice Information

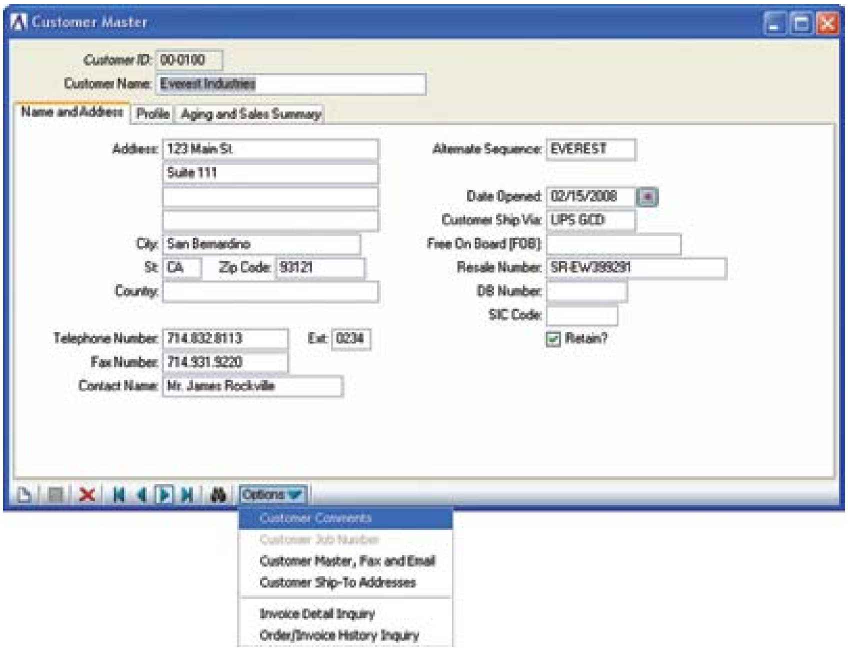
Distribution Code:
Payment Group:
Terms Code:
[X] Hold For Manual Release?
Invoice Date:
Accounting Date:
Due Date:
Discount Date:
Reference:
Memo:
Total Inv/Adj:
Net Invoice:
Discount:
Retention:
Dist Bal:

Account	Description	Misc.	Amount	Units

Figure 2. Header and detail tables easily linked into a one-to-many arrangement based on key structure

continued...

A detail window table is set up in much the same way, but when invoked from the main form through an options menu, it comes up as a separate modal form in Barista. **Figure 3** shows this structure in the Customer Maintenance form.



Customer Master

Customer ID:	00-0100
Customer Name:	Everest Industries

Name and Address | Profile | Aging and Sales Summary

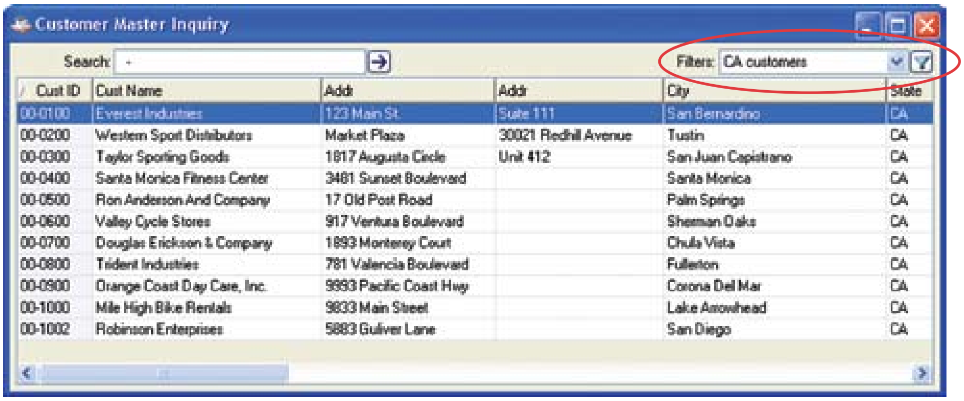
Address:	123 Main St	Alternate Sequence:	EVEREST
	Suite 111		
		Date Opened:	02/15/2008
		Customer Ship Via:	UPS GCD
City:	San Bernardino	Free On Board (FOB):	
St:	CA Zip Code: 93121	Resale Number:	SR-EW399291
Country:		DB Number:	
		SIC Code:	
Telephone Number:	714.832.8113 Ext: 0234	☑ Retain?	
Fax Number:	714.931.9220		
Contact Name:	Mr. James Rockville		

Options
Customer Comments
Customer Job Number
Customer Master, Fax and Email
Customer Ship-To Addresses
Invoice Detail Inquiry
Order/Invoice History Inquiry

Figure 3. Customer Master form with an Options Menu containing several detail window tables

Barista's Inquiry Subsystem

One of the biggest time savers in Barista is the inquiry subsystem. Barista inquiries make use of the BASIS SQL engine and any table-bound form or field has access to the inquiry subsystem. Legacy AddonSoftware contained an extensive set of field and record lookups, all of them the result of code written by a developer. By using Barista to develop the new AddonSoftware forms, developers were able to leave behind all of that old code. Once they defined a table, the running form could query the underlying table(s) for either the form or fields on the form, sort and filter, and even save often-used filters. **Figure 4** illustrates a saved filter for the Customer form that shows only those customers with California addresses. Users can even copy and paste information in the inquiry grid to other applications, such as a word processor or spreadsheet.



Customer Master Inquiry

Search: -

Filters: CA customers

Cust ID	Cust Name	Addr	Addr	City	State
00-0100	Everest Industries	123 Main St	Suite 111	San Bernardino	CA
00-0200	Western Sport Distributors	Market Plaza	30021 Redhill Avenue	Tustin	CA
00-0300	Taylor Sporting Goods	1817 Augusta Circle	Unit 412	San Juan Capistrano	CA
00-0400	Santa Monica Fitness Center	3481 Sunset Boulevard		Santa Monica	CA
00-0500	Ron Anderson And Company	17 Old Post Road		Palm Springs	CA
00-0600	Valley Cycle Stores	917 Ventura Boulevard		Sherman Oaks	CA
00-0700	Douglas Erickson & Company	1893 Monterey Court		Chula Vista	CA
00-0800	Trident Industries	781 Valencia Boulevard		Fullerton	CA
00-0900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy		Corona Del Mar	CA
00-1000	Mile High Bike Rentals	9833 Main Street		Lake Arrowhead	CA
00-1002	Robinson Enterprises	5883 Gulliver Lane		San Diego	CA

Figure 4. Quickly get to data with stored filters

The inquiry system, in conjunction with Barista's Document Output utility, also eliminates developer code by automatically generating listings of the various application code tables. In legacy AddonSoftware, there were separate programs to print the content of these kinds of tables. None of that code is necessary when utilizing Barista.

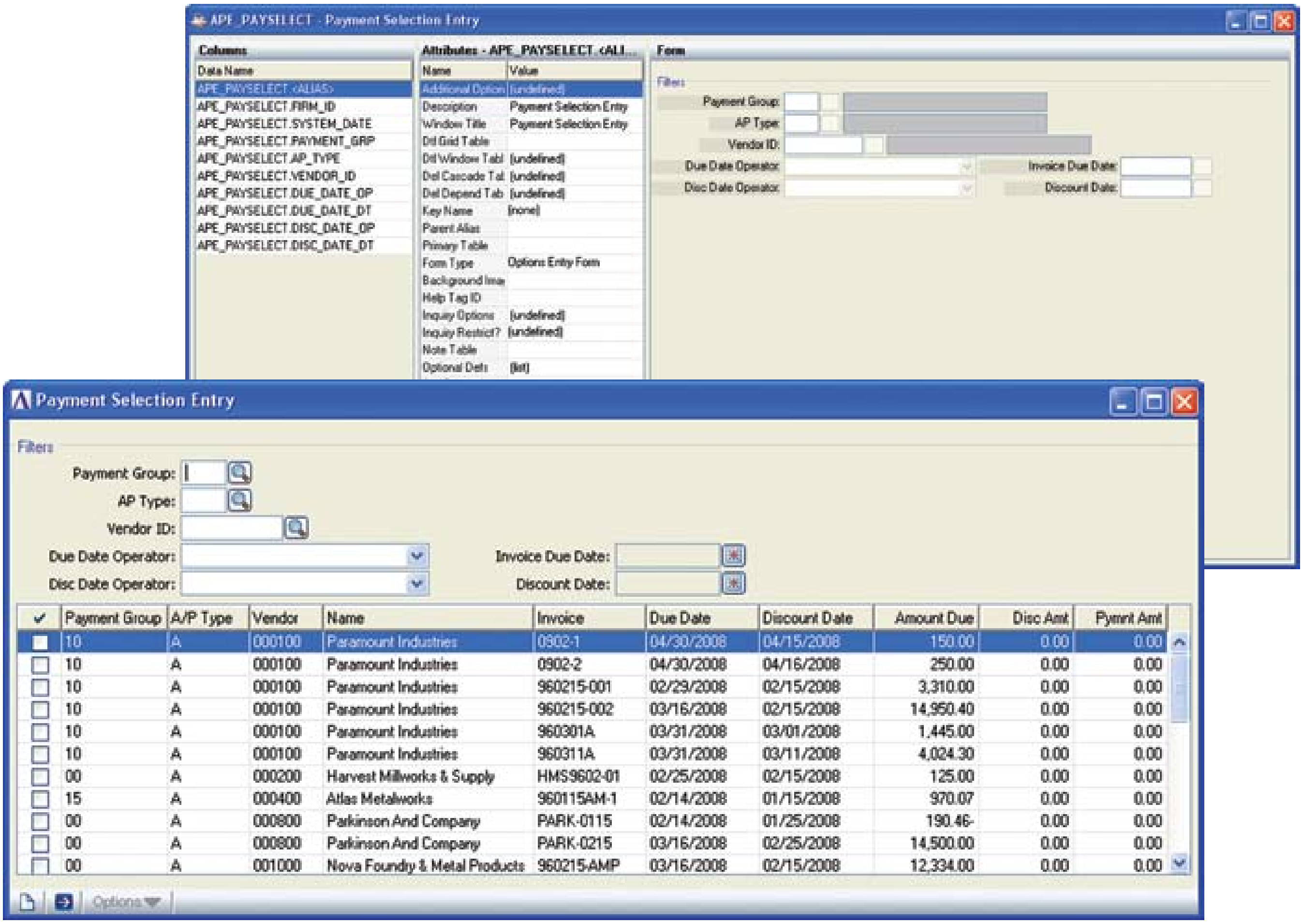
Barista Callpoints

Wondering how to implement old code? Whether the form is simple or complex, "you gotta get under the hood at some point," as they say. That is where Barista callpoints come into play. Barista callpoint code provides a gateway to supplement the Barista application with all that BBj and Java have to offer, thereby delivering flexibility to the framework's structure.

continued...

Barista callpoints are a series of event "hooks" associated with a Barista form, where a developer may want to place custom code. A few of the commonly used callpoints are "Before Form Show," "After Display," "Before Write," and "After Validation." The names themselves suggest their use. For example, inserting code in the "Before Write" callpoint of an invoice entry program would be a good way to make sure that amounts in the detail lines match the total invoice amount.

Like other applications, AddonSoftware contains many forms that are fairly simple and a few that are quite complex. Using Barista, the developers created many of the simpler forms in very little time, often with just a small amount of callpoint code for additional validation, parameter checking, etc. Some forms, however, needed functionality that is more complex. For example, the form in which the user selects invoices for payment (**Figure 5**) uses a combination of Barista-defined elements and a custom BBjGrid. Callpoint code handles creating and filling the grid, filtering the grid contents, setting callbacks and processing grid events, and writing the final grid results to a payment selection table. Callpoints provide the developer with all of the functionality of BBj, including BBj Custom Objects and Java classes. Consider incorporating other graphical tools into forms – a Java spinner, maybe – or launch the user's e-mail client or browser. Barista callpoints provide the portal for all of the additional code.


APE_PAYSELECT - Payment Selection Entry
Columns
Attributes - APE_PAYSELECT <ALI...
Form

Data Name
APE_PAYSELECT <ALIAS>
APE_PAYSELECT.FIRM_ID
APE_PAYSELECT.SYSTEM_DATE
APE_PAYSELECT.PAYMENT_GRP
APE_PAYSELECT.AP_TYPE
APE_PAYSELECT.VENDOR_ID
APE_PAYSELECT.DUE_DATE_OP
APE_PAYSELECT.DUE_DATE_DT
APE_PAYSELECT.DISC_DATE_OP
APE_PAYSELECT.DISC_DATE_DT

Name	Value
Additional Option	[undefined]
Description	Payment Selection Entry
Window Title	Payment Selection Entry
Dtl Grid Table	
Dtl Window Tabl	[undefined]
Del Cascade Tal	[undefined]
Del Depend Tab	[undefined]
Key Name	[none]
Parent Alias	
Primary Table	
Form Type	Options Entry Form
Background Ima	
Help Tag ID	
Inquiry Options	[undefined]
Inquiry Restrict?	[undefined]
Note Table	
Optional Dets	[list]

✓	Payment Group	A/P Type	Vendor	Name	Invoice	Due Date	Discount Date	Amount Due	Disc Amt	Pymnt Amt
□	10	A	000100	Paramount Industries	0902-1	04/30/2008	04/15/2008	150.00	0.00	0.00
□	10	A	000100	Paramount Industries	0902-2	04/30/2008	04/16/2008	250.00	0.00	0.00
□	10	A	000100	Paramount Industries	960215-001	02/29/2008	02/15/2008	3,310.00	0.00	0.00
□	10	A	000100	Paramount Industries	960215-002	03/16/2008	02/15/2008	14,950.40	0.00	0.00
□	10	A	000100	Paramount Industries	960301A	03/31/2008	03/01/2008	1,445.00	0.00	0.00
□	10	A	000100	Paramount Industries	960311A	03/31/2008	03/11/2008	4,024.30	0.00	0.00
□	00	A	000200	Harvest Millworks & Supply	HMS9602-01	02/25/2008	02/15/2008	125.00	0.00	0.00
□	15	A	000400	Atlas Metalworks	960115AM-1	02/14/2008	01/15/2008	970.07	0.00	0.00
□	00	A	000800	Parkinson And Company	PARK-0115	02/14/2008	01/25/2008	190.46-	0.00	0.00
□	00	A	000800	Parkinson And Company	PARK-0215	03/16/2008	02/25/2008	14,500.00	0.00	0.00
□	00	A	001000	Nova Foundry & Metal Products	960215-AMP	03/16/2008	02/15/2008	12,334.00	0.00	0.00

Figure 5. The grid shows on the running form, built and managed entirely with callpoint code

Going Totally Custom

Alternatively, developers can go completely outside the Barista form and callpoint structure to write their own custom BBj programs and then access them in Barista. These programs run seamlessly with the rest of the application inside the Barista MDI. The AddonSoftware Executive Summary is a case in point. Created with the BASIS IDE, it uses BBjTree, BBjGrid, and BBjChart controls to present graphical representations of various AddonSoftware data file compilations.

At the Back End

Codeport

The various back-end programs in legacy AddonSoftware were in need of some rejuvenating. Some of the changes were necessary and others, while not essential, transformed the code into a more modern, structured format, facilitating future maintenance tasks. The Codeport program is an AddonSoftware-specific utility that converts legacy AddonSoftware programs into source files compatible with Barista. Codeport takes advantage of the fact that this legacy code is highly standardized so it is possible to reliably locate and alter or replace code. What Codeport cannot do automatically, developers can do manually with the BASIS IDE.

Continue on page 44

Starwood's Web Service Shines Brighter

By Susan Darling

Imagine you are the leading developer and operator of high quality vacation ownership resorts. Each day, thousands of owners depend on your agents to manage their personal data, provide loan pay off values, make credit card transactions, and even schedule services for their properties. Your PRO/5® applications reliably provide these services, but only to your in-house agents. With a growing number of Internet services across all industries, you too would like to offer your services to your owners. What is the best approach to providing your owners with access to PRO/5 data and business logic through the Web?

Starwood Vacation Ownership (SVO) is that developer. Matt Lewis, SVO's Manager of Architecture explained, "We needed a secure, standards-based solution that we could expose to our external Web sites. Access to our data beyond the functionality of JDBC and ODBC drivers was critical - we wanted to access the business logic that already existed in our PRO/5 system."

Integration Environment Stats

Run BBj 5.0, PRO/5 4.10, Java 1.5, and Solaris 10 on two dual-processor SunSPARCs

★

200 users run Web services for internal applications

★

3,000-5,000 external users a day talk to a back-end license of 800 users

★

200,000 transactions per week averaging $1.5-2 million (over $100 million annually)

Considering all the options, the answer was clear. Lewis led his team to a BBj® solution using Web services that would do all that and more.

StarCentral Integration

Starwood Vacation Ownership had an existing owner-based Web site, StarCentral, that made use of batch-loaded data and transaction requests that agents later processed manually. SVO wanted to make this owner data real time and automate these transactions. In just over a year, the internal development team made up of PRO/5, BBj, and Java developers, was able to make a majority of data and transactions real time - owner data, loan payments, loan balance, and homeowners' association (HOA) balances, and payments. The team utilized BBj's capabilities to access PRO/5 application logic and data to provide real-time Web services that talked with their existing PRO/5 systems. New owners now simply log in through an authentication process, and then in real time, access Starwood's back-end system. Owners can now review their contract data, list all their ownerships, view their status and make loan and HOA payments. StarCentral became the owners' command central where they make reservation requests, pay accounts, read owner news, communicate with Starwood, and get other valuable ownership information. Surfing through these options allow owners to manage their data needs easily and intuitively from their own unique computer configurations. StarCentral empowers owners and releases them to enjoy a new level of data freedom.

Shining Brighter

Recognizing that owners expected similar performance as elsewhere on the Web, Starwood spent extra effort speeding up the performance of the system and its integration with PRO/5. Proudly, most Web services calls take less than one second and credit card processing takes an average of six, which is quite acceptable to the owners.

The Web services solution also met Starwood's need to access the business logic in their PRO/5 application. For example, when calculating the mortgage payoff to the current day, owners and agents both rely on the business logic already embedded in Starwood's PRO/5 programs. Since this information is dynamically calculated and therefore not stored in the database, owners can run the same program over the Web that the agents run locally. Using the existing proven PRO/5 code base is a guarantee that both the owner and agent get the same real-time results. The savings in programming, testing, and deploying one code base is far reaching.

Trouble*SHOOTING STAR*Central

As the volume of transactions increases, so does the need for more refined and effective troubleshooting tools. One significant step was their addition of globally unique identifiers for each transaction. With this in place, Starwood

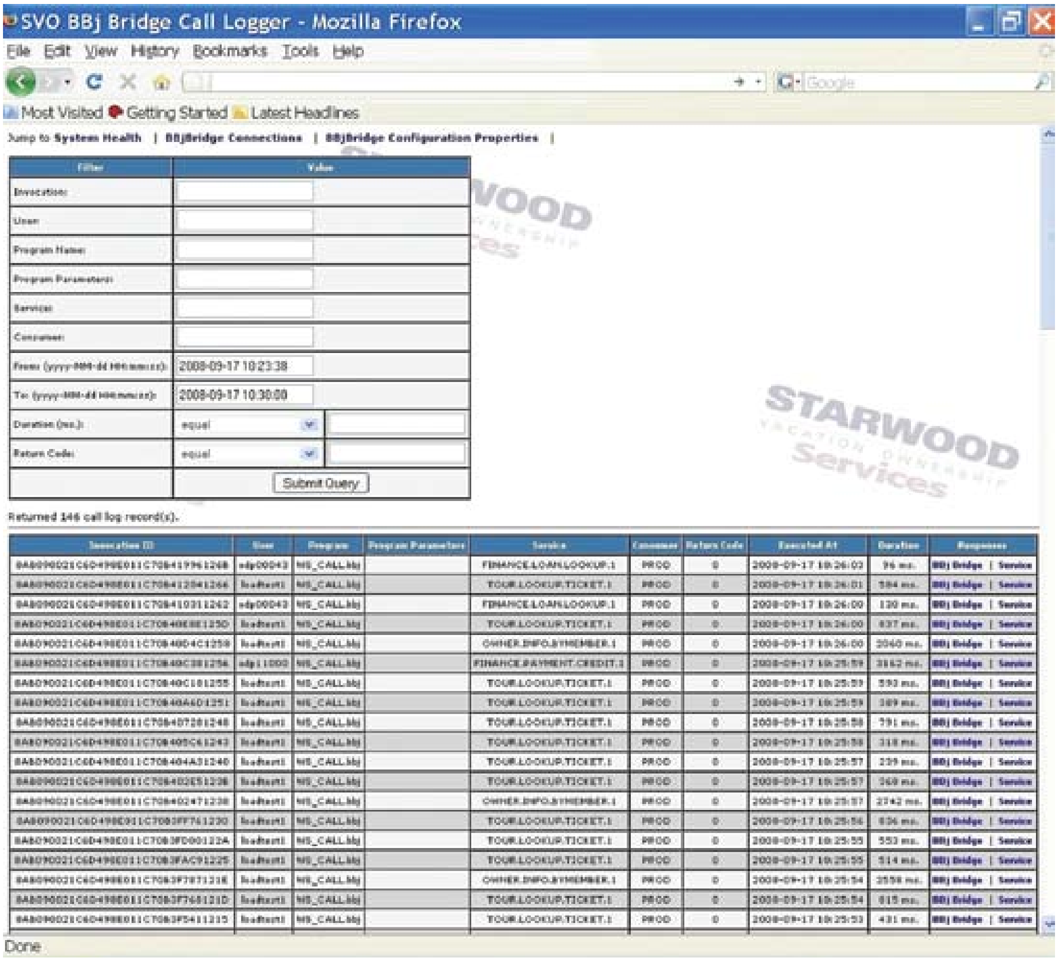
continued...

Matt Lewis, *Development Manager at Starwood Vacation Ownership, leads the development team of PRO/5, BBj, and Java developers. Matt holds a degree in Computer Science from The Citadel - Military College of South Carolina and completed some graduate courses at the University of California at Berkeley and the University of New Delhi in India.*

In the fast-growing segment of vacation ownership, Starwood Vacation Ownership *is setting the benchmark, consistently producing quality products in some of the world's most sought-after destinations. Their focused vision has established a strong foundation for growth and future successes.*

Starwood Vacation Ownership is one of the premier developers and operators of high-quality vacation ownership resorts. The company is a wholly-owned subsidiary of Starwood Hotels and Resorts Worldwide, Inc. (NYSE: HOT) and currently operates vacation ownership resorts under the Westin, Sheraton, and St. Regis brands in some of the world's most desirable destinations including Hawaii, Colorado and the Caribbean. Starwood, headquartered in Orlando, Florida, has 18 resorts and more than 4,300 employees. Visit www.starwoodvacationownership.com.

Susan Darling
Technical/Marketing Writer


SVO BBj Bridge Call Logger - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Google
Most Visited Getting Started Latest Headlines
Done

Jump to System Health | BBjBridge Connections | BBjBridge Configuration Properties |

Filter	Value	
Invocation:		
User:		
Program Name:		
Program Parameters:		
Service:		
Consumer:		
From: (yyyy-MM-dd HH:mm:ss):	2008-09-17 10:23:38	
To: (yyyy-MM-dd HH:mm:ss):	2008-09-17 10:30:00	
Duration (ms.):	equal	
Return Code:	equal	
	Submit Query	

Returned 146 call log record(s).

Invocation ID	User	Program	Program Parameters	Service	Consumer	Return Code	Executed At	Duration	Responses
8A8090021C6D498E011C70B419961268	edp00043	WS_CALL.bbj		FINANCE.LOAN.LOOKUP.1	PROD	0	2008-09-17 10:26:03	96 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B412041266	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:26:01	584 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B410311262	edp00043	WS_CALL.bbj		FINANCE.LOAN.LOOKUP.1	PROD	0	2008-09-17 10:26:00	120 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B40E8E1250	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:26:00	437 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B40D4C1258	loadtest1	WS_CALL.bbj		OWNER.INFO.BYMEMBER.1	PROD	0	2008-09-17 10:26:00	2040 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B40C381256	edp11000	WS_CALL.bbj		FINANCE.PAYMENT.CREDIT.1	PROD	0	2008-09-17 10:25:59	3162 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B40C181255	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:59	593 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B40A6D1251	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:59	389 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B407281248	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:58	791 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B405C61243	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:58	318 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B404A31240	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:57	239 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B402E51238	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:57	369 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B402471238	loadtest1	WS_CALL.bbj		OWNER.INFO.BYMEMBER.1	PROD	0	2008-09-17 10:25:57	2742 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3FF761230	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:56	636 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3FD0D122A	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:55	553 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3FAC91225	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:55	514 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3F787121E	loadtest1	WS_CALL.bbj		OWNER.INFO.BYMEMBER.1	PROD	0	2008-09-17 10:25:54	2558 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3F768121D	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:54	615 ms.	BBj Bridge | Service
8A8090021C6D498E011C70B3F5411215	loadtest1	WS_CALL.bbj		TOUR.LOOKUP.TICKET.1	PROD	0	2008-09-17 10:25:53	431 ms.	BBj Bridge | Service

Figure 1. BBj Bridge Call Logger

can replay any transaction at any point in time to see the trouble spot in action. Another step was to add effective dates. Their PRO/5 system has many rules that pertain to certain times of the month or of the year so they use effective dates to test those rules in their services environment. Refining their reporting and debugging tools is a high priority in an effort to improve their troubleshooting and problem solving skills.

Stargazing

StarCentral is a highly available, load-balanced environment. Currently, they average 3,000-5,000 log ins per day and take credit card payment amounts ranging from $29 to $80,000. Load balancing allows them to present one system or URL to the user for both machines and the system routes the traffic to the available machine based on the load. It also has a fail-over feature to re-route all traffic in the event of a system failure.

To keep a careful eye on the system, Starwood built in a real-time monitoring system. In **Figure 1**, the BBj Bridge Call Logger, reports real-time queries of transactions based on the user, transaction ID, name of service, to/from dates, duration time of transaction, and the return code. Starwood also relies on JMeter, a desktop application by Apache, to report performance-testing data vital for further analysis. They constantly load and performance-test their environment to the upper capacity of 200 concurrent transactions per second – far beyond today's needs.

Star Struck

"We are pretty happy with the stability of BBj 5.0. We do have plans though to go to 8.21 after we run it 24x7 for a few months in QA and development," says Lewis. As their load increases, BBj's scalability easily allows them to add more hardware.

Starwood's BBj-powered solution processes more than $100 million per year, illustrating the high value of BBj's scalability. Even if your current or future Web services application does not propel you into this stratosphere, there are strong benefits that you can take advantage of today.

Join Starwood and stretch your application to achieve its greatest potential through BASIS.

Read more about Web services in earlier issues of the *BASIS International Advantage*

BBj and Web Services
www.basis.com/advantage/mag-v7n1/webservices.html

Writing a Web Service in BBj
www.basis.com/advantage/mag-v7n2/webservice.html

The QA Memos Web Service
www.basis.com/advantage/mag-v9n2/memos.html

From the BASIS DD to a Barista App in a Flash

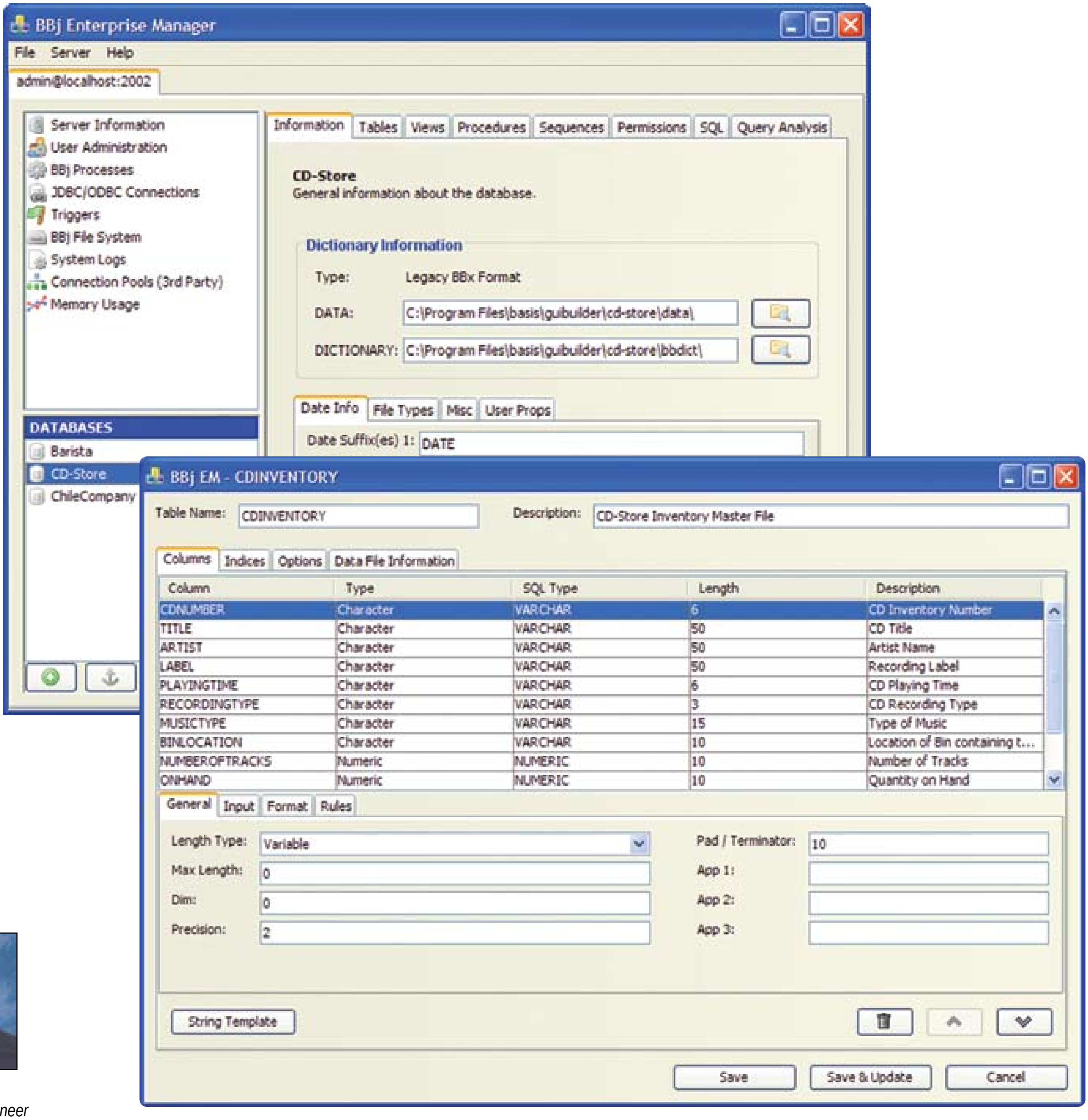
By Jim Douglas

If you have a BASIS Data Dictionary definition of your data files, you are well on your way to turning your older GUI or CUI application into a modern GUI application. Because Barista is a data dictionary-based development tool, you can quickly use the Barista Data Dictionary Import Tool to get a head start on your development efforts.

We'll show you how to:

✓ Import the Data Dictionary
✓ Build a file maintenance form
✓ Automatically add a standard report
✓ Create a menu
✓ Add an existing report to the menu
✓ Test-drive the form
✓ Exercise the built-in SQL inquiry feature
✓ Produce output in a PDF format
✓ E-mail the result from within Barista

In the space of a few minutes we'll go from this:



CD-Store
General information about the database.

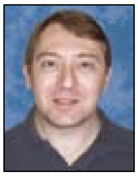
Type: Legacy BBx Format
DATA: C:\Program Files\basis\guibuilder\cd-store\data\
DICTIONARY: C:\Program Files\basis\guibuilder\cd-store\bbdict\
Date Suffix(es) 1: DATE

Table Name: CDINVENTORY
Description: CD-Store Inventory Master File

Column	Type	SQL Type	Length	Description
CDNUMBER	Character	VARCHAR	6	CD Inventory Number
TITLE	Character	VARCHAR	50	CD Title
ARTIST	Character	VARCHAR	50	Artist Name
LABEL	Character	VARCHAR	50	Recording Label
PLAYINGTIME	Character	VARCHAR	6	CD Playing Time
RECORDINGTYPE	Character	VARCHAR	3	CD Recording Type
MUSICTYPE	Character	VARCHAR	15	Type of Music
BINLOCATION	Character	VARCHAR	10	Location of Bin containing t...
NUMBEROFTRACKS	Numeric	NUMERIC	10	Number of Tracks
ONHAND	Numeric	NUMERIC	10	Quantity on Hand

Length Type: Variable
Max Length: 0
Dim: 0
Precision: 2
Pad / Terminator: 10
App 1:
App 2:
App 3:

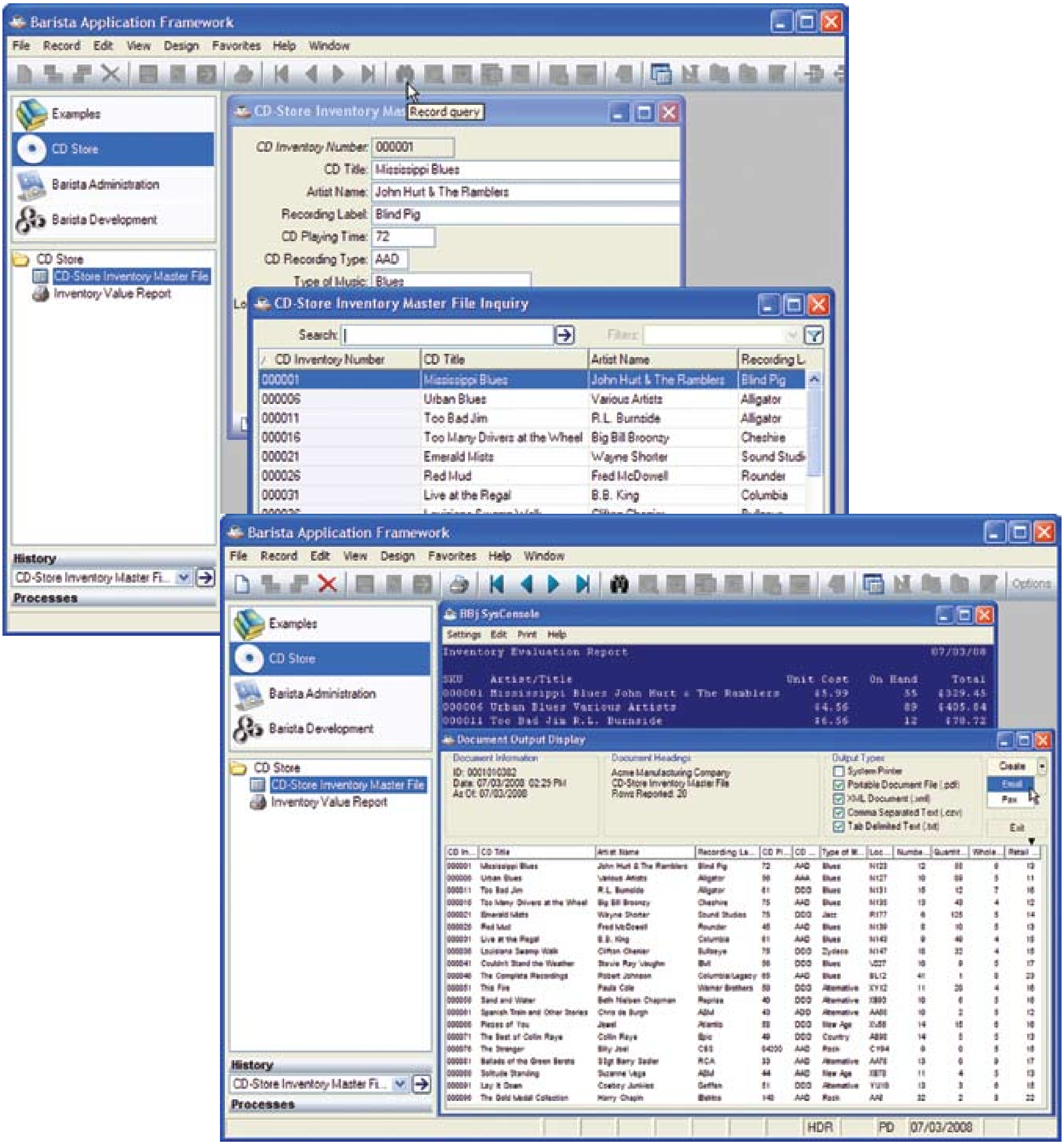
Figure 1. Original database in need of a GUI modern front end



Jim Douglas
Software Engineer
Contractor

continued...

To this:



CD-Store Inventory Master File

CD Inventory Number: 000001
CD Title: Mississippi Blues
Artist Name: John Hurt & The Ramblers
Recording Label: Blind Pig
CD Playing Time: 72
CD Recording Type: AAD
Type of Music: Blues

CD-Store Inventory Master File Inquiry

Search:
Filters:

CD Inventory Number	CD Title	Artist Name	Recording L...
000001	Mississippi Blues	John Hurt & The Ramblers	Blind Pig
000006	Urban Blues	Various Artists	Alligator
000011	Too Bad Jim	R.L. Burnside	Alligator
000016	Too Many Drivers at the Wheel	Big Bill Broonzy	Cheshire
000021	Emerald Mists	Wayne Shorter	Sound Studi
000026	Red Mud	Fred McDowell	Rounder
000031	Live at the Regal	B.B. King	Columbia

BBj SysConsole

```
Inventory Evaluation Report                                         07/03/08

SKU    Artist/Title                                   Unit Cost   On Hand     Total
000001 Mississippi Blues John Hurt & The Ramblers         $5.99        55   $329.45
000006 Urban Blues Various Artists                        $4.56        89   $405.84
000011 Too Bad Jim R.L. Burnside                          $6.56        12    $78.72
```

Document Output Display

Document Information
ID: 0001010382
Date: 07/03/2008 02:29 PM
As Of: 07/03/2008

Document Headings
Acme Manufacturing Company
CD-Store Inventory Master File
Rows Reported: 20

Output Types
☐ System Printer
☑ Portable Document File (.pdf)
☑ XML Document (.xml)
☑ Comma Separated Text (.csv)
☑ Tab Delimited Text (.txt)

CD In...	CD Title	Artist Name	Recording La...	CD Pl...	CD ...	Type of M...	Loc...	Numbe...	Quantit...	Whole...	Retail ...
000001	Mississippi Blues	John Hurt & The Ramblers	Blind Pig	72	AAD	Blues	N123	12	55	6	13
000006	Urban Blues	Various Artists	Alligator	56	AAA	Blues	N127	10	89	5	11
000011	Too Bad Jim	R.L. Burnside	Alligator	61	DDD	Blues	N131	15	12	7	16
000016	Too Many Drivers at the Wheel	Big Bill Broonzy	Cheshire	75	AAD	Blues	N135	13	48	4	12
000021	Emerald Mists	Wayne Shorter	Sound Studios	75	DDD	Jazz	R177	8	125	5	14
000026	Red Mud	Fred McDowell	Rounder	45	AAD	Blues	N139	8	10	5	13
000031	Live at the Regal	B.B. King	Columbia	61	AAD	Blues	N143	9	40	4	15
000036	Louisiana Swamp Walk	Clifton Chenier	Bullseye	78	DDD	Zydeco	N147	18	32	4	15
000041	Couldn't Stand the Weather	Stevie Ray Vaughn	[illegible]	56	DDD	Blues	V227	16	9	5	17
000046	The Complete Recordings	Robert Johnson	Columbia/Legacy	65	AAD	Blues	BL12	41	1	8	23
000051	This Fire	Paula Cole	Warner Brothers	50	DDD	Alternative	XY12	11	20	4	16
000056	Sand and Water	Beth Nielsen Chapman	Reprise	40	DDD	Alternative	X800	10	6	5	16
000061	Spanish Train and Other Stories	Chris de Burgh	A&M	43	ADD	Alternative	AA80	10	2	5	12
000066	Pieces of You	Jewel	Atlantic	55	DDD	New Age	XJ56	14	15	6	16
000071	The Best of Collin Raye	Collin Raye	Epic	49	DDD	Country	AB90	14	5	5	13
000076	The Stranger	Billy Joel	CBS	64200	AAD	Rock	CY84	9	6	5	16
000081	Ballads of the Green Berets	SSgt Barry Sadler	RCA	33	AAD	Alternative	AA78	13	6	9	17
000086	Solitude Standing	Suzanne Vega	A&M	44	AAD	New Age	XB78	11	4	5	13
000091	Lay It Down	Cowboy Junkies	Geffen	51	DDD	Alternative	YU18	13	3	6	16
000096	The Gold Medal Collection	Harry Chapin	Elektra	148	AAD	Rock	AA8	32	2	8	22

Figure 2. Newly created application using Barista

Conclusion

Your current BASIS Data Dictionary is perfect raw material for your first Barista-brewed application. Barista facilitates the extremely rapid creation and modernization of business applications. Cut out the enormously time-consuming GUI design and management tasks normally associated with the creation of a GUI application. Add luster and shine to your trusted and proven solution and leverage BBj's access to Java libraries, such as faxing modules, to extend your applications functionality.

Take a few minutes to follow the steps laid out online and transform your venerable application to a tasty eye-catching delight.

There is more!

For a hands-on example of creating a modern GUI application, follow along at
www.basis.com/advantage/mag-v12n1/barista2.pdf

Dishing up Forms

By Robert Del Prete

The BBjForm object, combined with the getBBjPDFForm() method and the BBjClientFileSystem, make generating and distributing forms fast, easy, and accurate. This article reviews the BBj® Objects and methods that give developers all the tools they need to dish up even the most complicated forms.

When it all Began

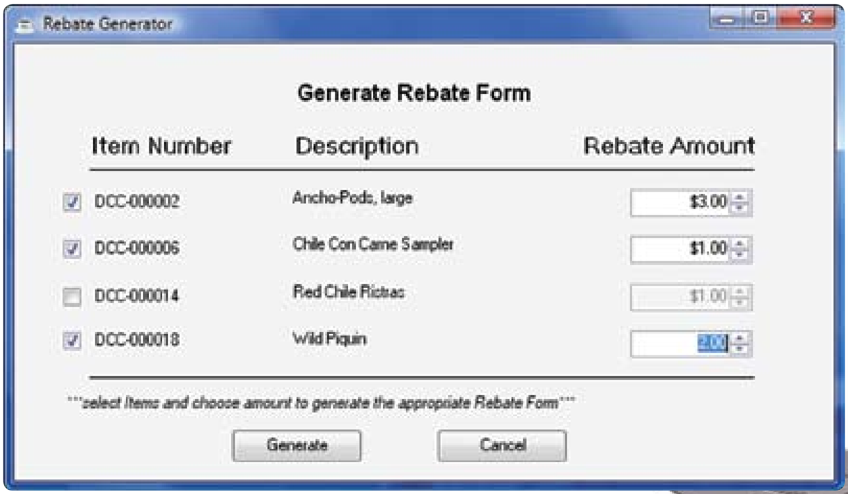
BBj 4.0 introduced the BBjForm object to simplify document and report creation within BBj and to provide advanced formatting capability and greater access to printers. Creation of reports with complex layout requirements and multiple font types and images became much easier for BBj developers to create.

The getBBjPDFForm() method, introduced in BBj 6.0, allows the creation of documents in Portable Document Format (PDF) format. Created by Adobe in 1993, PDF files include text, images, multiple fonts, and graphics, and are viewable by multiple applications on different operating systems. With so many options for creating and viewing documents, PDF gives developers a rich and widely accepted environment in which to create their forms.

To complement these features, BBj 7.0 introduced the BBjClientFileSystem to provide developers with an API to access and manipulate files on the client without needing a BBjFileSystem server on the client.

The Utensils

The BBjClientFile and BBjForm objects combine to give developers a better structure for their printing routines and more effectively meet the needs of their customer base. The "Rebate Generator" demo† (shown in **Figure 1** and downloadable with BBj), utilizes these objects to allow BBj clients to display and/or print a PDF document. This program includes formatted text, bar code fonts, images, and graphics. It moves the PDF to the client using the BBjClientFileSystem and displays the PDF via a BBj multimedia viewer program created using client objects.



Generate Rebate Form

	Item Number	Description	Rebate Amount
☑	DCC-000002	Ancho-Pods, large	$3.00
☑	DCC-000006	Chile Con Carne Sampler	$1.00
□	DCC-000014	Red Chile Ristras	$1.00
☑	DCC-000018	Wild Piquin	3.00

select Items and choose amount to generate the appropriate Rebate Form

Figure 1. The "Rebate Generator" demo

Create the Form

Developers can use the getBBjPDFForm() method to obtain a BBjForm object and subsequently add the page object using createPage(). Then they simply need to add text and images to the page object to create the rebate form.

```
api!=BBjAPI()
pdfform$="pdfform.pdf"
doc!=api!.getBBjPDFForm(pdfform$,1)
page!=doc!.createPage()
```

*Robert Del Prete
Quality Assurance
Engineer*

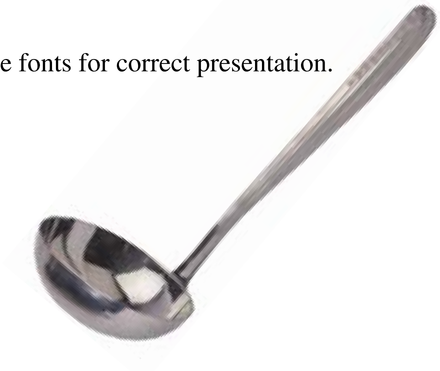
The getBBjPDFForm() method specifies the PDF image parameter to prevent complications on the client. This increases the size of the PDF image, but it also guarantees the end result displays exactly as rendered, regardless of

continued...

whether the barcode font exists on the target computer. If the PDF document needs editing, specify the document form of getBBjPDFForm() as follows:

```
Doc!=api!.getBBjPDFForm(pdfform$,0)
```

This form allows editing but relies on the client operating system to supply the fonts for correct presentation.



Add an Image

To add a new image to the page, specify the image file, location, and size.

```
dccccimage!=page!.newImage()
dccccimage!.setFile("DryCanyonLOGO.png")
dccccimage!.setPosition(new java.awt.Point(0,0))
dccccimage!.setWidth(566)
dccccimage!.setHeight(65)
```

Add a Graphic Line

To add a simple line graphic to the page, use newLine() and set the position, length, and thickness using BBjFormLine methods.

```
titleline1!=page!.newLine()
titleline1!.setRelative(0)
titleline1!.setPosition(new java.awt.Point(20,105))
titleline1!.setEndPoint(new java.awt.Point(580,105))
titleline1!.setThickness(1)
```

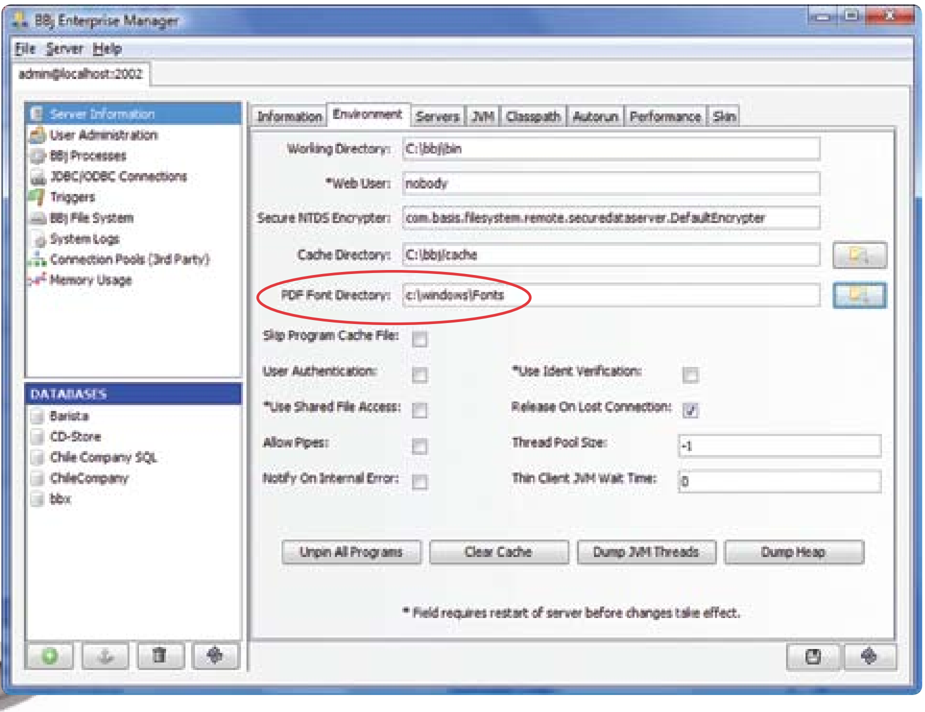
Add Text

Add text to the paragraph by specifying newParagraph() and set fonts, location, and size..

```
let smalltextfont!= new java.awt.Font("Arial",java.awt.Font.PLAIN,9)
qualifytext2!=page!.newParagraph()
qualifytext2!.setHorizontalFill(1)
qualifytext2!.setPosition(new java.awt.Point(15,445))
qualifytext2!.setText("To qualify for this rebate, please make sure you complete ALL of the following steps:")
qualifytext2!.setFont(smalltextfont!)
```

Notice the use of setFont() in the sample above. Any TrueType or PostScript font, including barcode fonts, is available to the PDF form provided the correct font location appears in the Enterprise Manager "PDF Font Directory" field as shown in **Figure 2**. Once this configuration is complete, this process of adding barcode fonts to the form is the same as changing the font on any other paragraph.

continued...

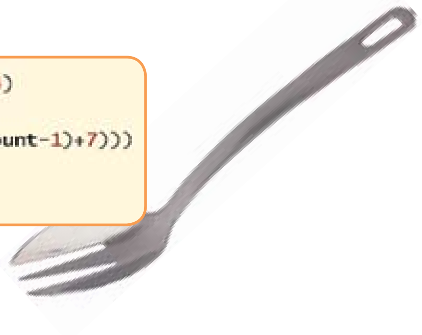


Working Directory: C:\bbj\bin
*Web User: nobody
Secure NTDS Encrypter: com.basis.filesystem.remote.securedataserver.DefaultEncrypter
Cache Directory: C:\bbj\cache
PDF Font Directory: c:\windows\Fonts
Skip Program Cache File: □
User Authentication: □
*Use Ident Verification: □
*Use Shared File Access: □
Release On Lost Connection: ☑
Allow Pipes: □
Thread Pool Size: -1
Notify On Internal Error: □
Thin Client JVM Wait Time: 0

* Field requires restart of server before changes take effect.

Figure 2. Configure PDF Font Directory in the Enterprise Manager

Use a Barcode Font



```
let barcodeFont!=new java.awt.Font("IDAutomationSXHC39M",java.awt.Font.PLAIN,14)
product1upc!=page!.newParagraph()
product1upc!.setHorizontalFill(1)
product1upc!.setPosition(new java.awt.Point(340,(positionVector!.getItem(itemcount-1)+7)))
product1upc!.setText("32789823")
product1upc!.setFont(barcodefont!)
page!.add(product1upc!)
```

Ready to Serve

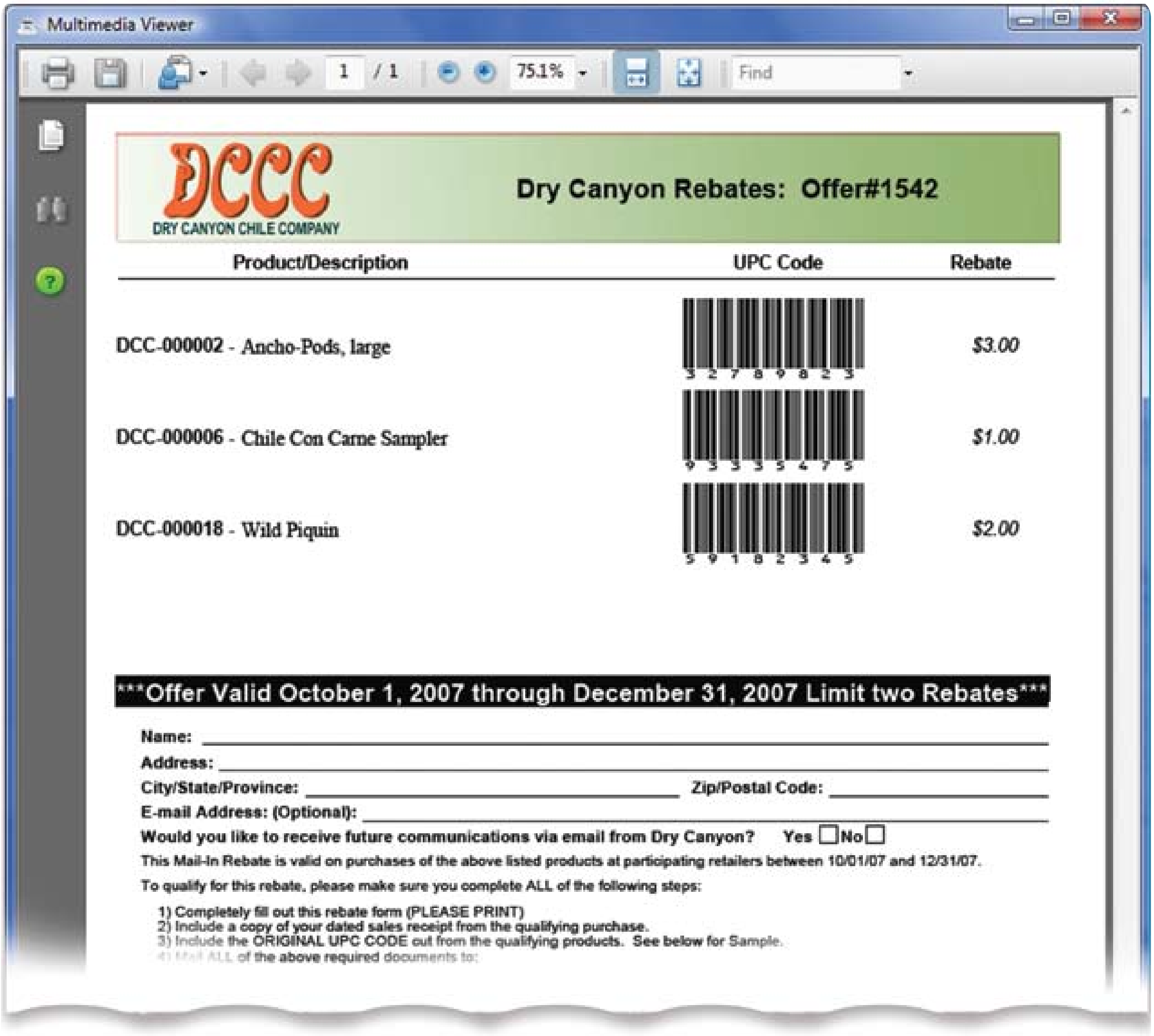
After adding text and images, render the document as a PDF image.

Distributing the form to the client is possible via the BBClientFileSystem. The copyToClient() method copies the newly created PDF file over to the client.

```
mcf! = api!.getThinClient().getClientFileSystem()
clientpdf$=FILESAVE("Save PDF File to Client",FILE_DIR$,"","",FILTER$)
clientpdf!=mcf!.getClientFile(clientpdf$)
clientpdf!.copyToClient(filepdf$)
```

The getClientFile() method retrieves the file on the remote client while the copyToClient() copies the file over to the client. The finished PDF form appears in **Figure 3**.

The "Multi-Media Viewer" demo† program uses client objects to display the PDF file on the client system. The end result is a PDF document created, shipped, and displayed on the client using BBj. The document can be viewed, printed, e-mailed, or saved for later use.



Dry Canyon Rebates: Offer#1542

Product/Description	UPC Code	Rebate
DCC-000002 - Ancho-Pods, large	32789823	$3.00
DCC-000006 - Chile Con Carne Sampler	93335475	$1.00
DCC-000018 - Wild Piquin	59182345	$2.00

Offer Valid October 1, 2007 through December 31, 2007 Limit two Rebates

Name: ____________________

Address: ____________________

City/State/Province: ____________________ Zip/Postal Code: ____________________

E-mail Address: (Optional): ____________________

Would you like to receive future communications via email from Dry Canyon? Yes □ No □

This Mail-In Rebate is valid on purchases of the above listed products at participating retailers between 10/01/07 and 12/31/07.

To qualify for this rebate, please make sure you complete ALL of the following steps:

1) Completely fill out this rebate form (PLEASE PRINT)
2) Include a copy of your dated sales receipt from the qualifying purchase.
3) Include the ORIGINAL UPC CODE cut from the qualifying products. See below for Sample.
4) Mail ALL of the above required documents to:

Figure 3. The resultant PDF of the "Rebate Generator" demo

continued...

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

Summary

Using the BBjForm object along with the BBjClientFileSystem gives developers a wide range of options for creating and deploying forms within their application. Remote clients can now view and retrieve complex forms generated on the server. Dishing up forms has never been easier!

Read more about PDF forms in earlier issues of the *BASIS International Advantage*

BBj Form Printing
www.basis.com/advantage/mag-v7n3/printing.html

PDF Now Also Means Perfectly Displayed Forms
www.basis.com/advantage/mag-v9n2/pdf.html

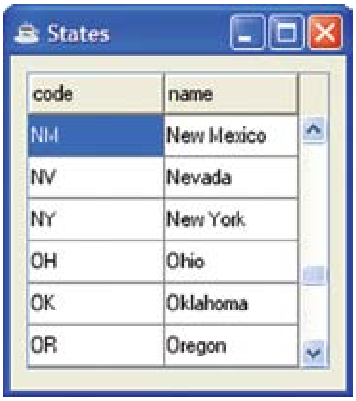
Desktop Data Delivered
www.basis.com/advantage/mag-v11n1/DesktopData01adv07_Links.pdf

† Download with BBj from www.basis.com/products/bbj/download.html and select "Demos" in the Optional File section. After completing the installation, select **BBj > Demos > LaunchDock** from the BASIS folder to run the demo.

Databound Grid – Data Abstraction Extended

By Jim Douglas

The BBjDataBoundGrid provides an easy way to display and edit data from a BASIS multikeyed file (MKEYED, XKEYED or VKEYED) or from any SQL-accessible database. It builds on recordsets and databound controls first introduced in BBj 4.0. Databound controls define the association between fields in the database and GUI controls on the screen. With the databound controls automatically transferring data between the screen and the database, the developer can focus on application details to build data-oriented applications more quickly and easily than ever before.



code	name
NM	New Mexico
NV	Nevada
NY	New York
OH	Ohio
OK	Oklahoma
OR	Oregon

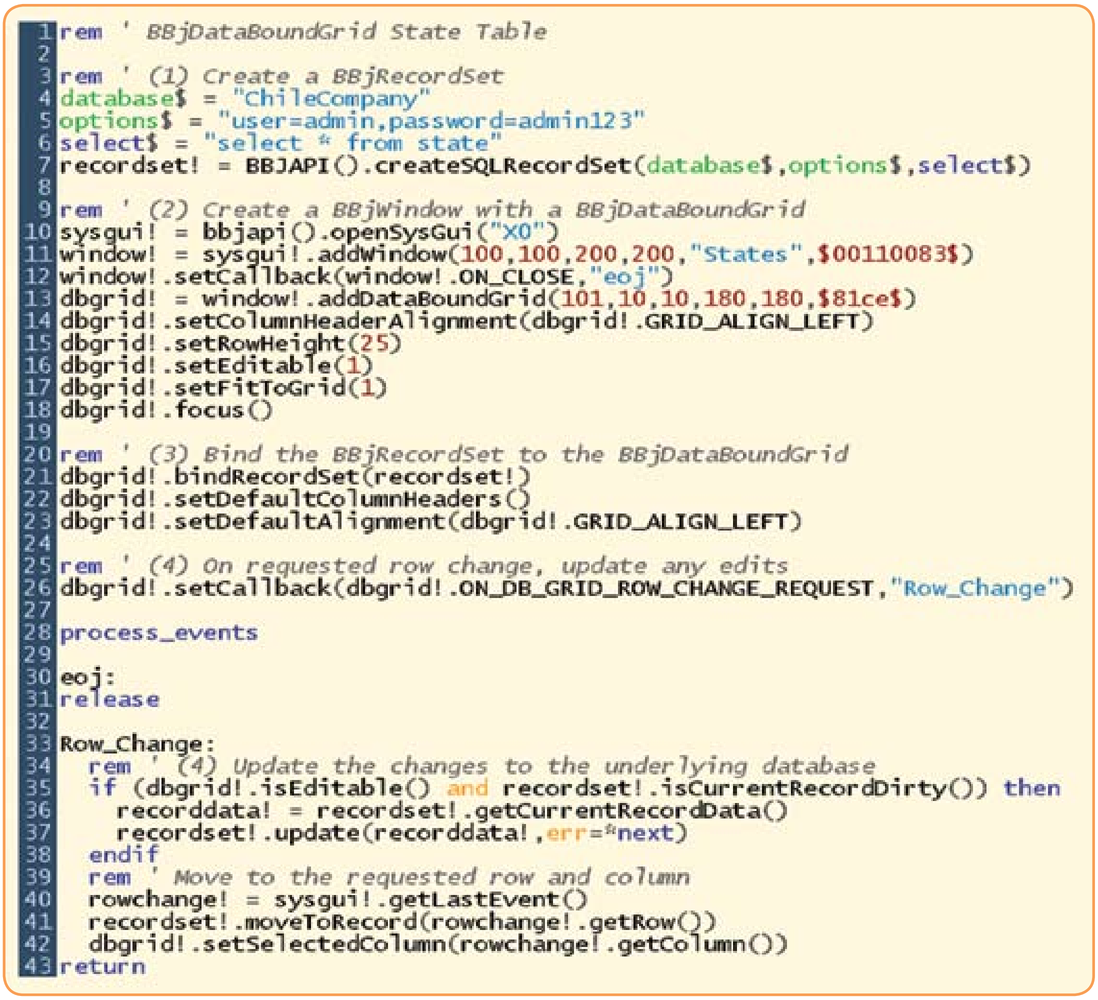
Figure 1. BBjDataBoundGrid state table

File Maintenance Sample

Figure 1 shows a typical grid-based maintenance screen for viewing and editing state codes.

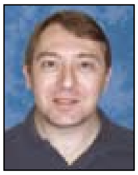
Figure 2 shows the complete source that produced this maintenance screen.

continued...



```
rem ' BBjDataBoundGrid State Table

rem ' (1) Create a BBjRecordSet
database$ = "ChileCompany"
options$ = "user=admin,password=admin123"
select$ = "select * from state"
recordset! = BBJAPI().createSQLRecordSet(database$,options$,select$)

rem ' (2) Create a BBjWindow with a BBjDataBoundGrid
sysgui! = bbjapi().openSysGui("X0")
window! = sysgui!.addWindow(100,100,200,200,"States",$00110083$)
window!.setCallback(window!.ON_CLOSE,"eoj")
dbgrid! = window!.addDataBoundGrid(101,10,10,180,180,$81ce$)
dbgrid!.setColumnHeaderAlignment(dbgrid!.GRID_ALIGN_LEFT)
dbgrid!.setRowHeight(25)
dbgrid!.setEditable(1)
dbgrid!.setFitToGrid(1)
dbgrid!.focus()

rem ' (3) Bind the BBjRecordSet to the BBjDataBoundGrid
dbgrid!.bindRecordSet(recordset!)
dbgrid!.setDefaultColumnHeaders()
dbgrid!.setDefaultAlignment(dbgrid!.GRID_ALIGN_LEFT)

rem ' (4) On requested row change, update any edits
dbgrid!.setCallback(dbgrid!.ON_DB_GRID_ROW_CHANGE_REQUEST,"Row_Change")

process_events

eoj:
release

Row_Change:
  rem ' (4) Update the changes to the underlying database
  if (dbgrid!.isEditable() and recordset!.isCurrentRecordDirty()) then
    recorddata! = recordset!.getCurrentRecordData()
    recordset!.update(recorddata!,err=*next)
  endif
  rem ' Move to the requested row and column
  rowchange! = sysgui!.getLastEvent()
  recordset!.moveToRecord(rowchange!.getRow())
  dbgrid!.setSelectedColumn(rowchange!.getColumn())
return
```

Figure 2. BBjDataBoundGrid state table – listing `(State.src)`



Jim Douglas
Software Engineer
Contractor

The main sections of this program are:

`(1) Create a BBjRecordSet`
Uses an SQLRecordSet that maps to the state table of the ChileCompany database, which is an MKEYED file with an associated entry in the Data Dictionary. Recordsets can also be associated with BBj multikeyed files (FileRecordSet) via a string template or with memory-based data structures (MemoryRecordSet) that might have been constructed from READ or READ RECORD statements applied to non-normalized data files.

`(2) Create a BBjWindow with a BBjDataBoundGrid`

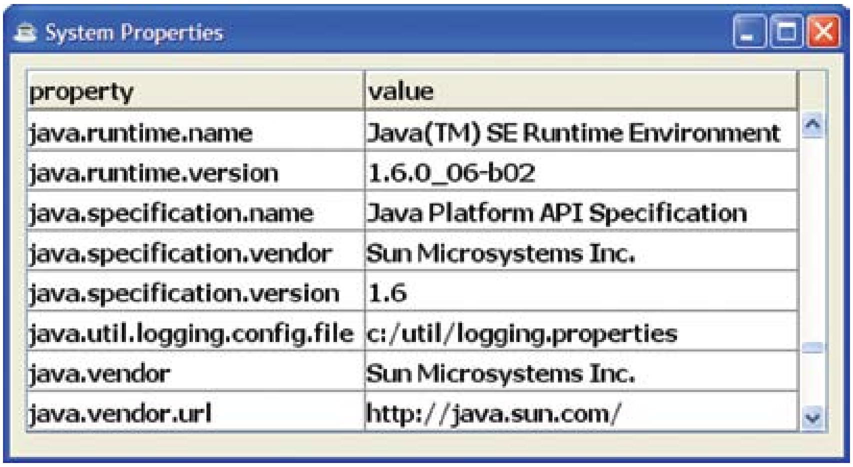
`(3) Bind the BBjRecordSet to the BBjDataBoundGrid`
Associates the databound grid to the recordset and creates an ongoing two-way linkage between the grid and the underlying data structure.

`(4) Update the changes to the underlying database`
When the user attempts to move to a new row, BBj fires a BBjDBGridRowChangeRequestEvent for which the programmer may choose how to respond. This sample calls BBjRecordSet::isCurrentRecordDirty() to see if any changes were made; if so, it writes the changed record to the underlying database. Finally, it moves to the requested row and column.

To see the two-way linkage in action, run two copies of `State.src` at the same time. Change one of the state names on one screen and move away from that row to commit the change to the database. When navigating again to that row on the other screen, the value updates automatically to reflect the change.

Property Sheet Sample

Databound grids are a convenient way to display any table-oriented data. **Figure 3** displays a typical property sheet that shows current Java system properties. The source code list in **Figure 4** uses a memory-based databound grid for the property sheet sample.



property	value
java.runtime.name	Java(TM) SE Runtime Environment
java.runtime.version	1.6.0_06-b02
java.specification.name	Java Platform API Specification
java.specification.vendor	Sun Microsystems Inc.
java.specification.version	1.6
java.util.logging.config.file	c:/util/logging.properties
java.vendor	Sun Microsystems Inc.
java.vendor.url	http://java.sun.com/

Figure 3. BBjDataBoundGrid property sheet

The main sections of this program are similar to the first sample:

`(1) Create a BBjRecordSet`
Uses a MemoryRecordSet of properties and corresponding values.

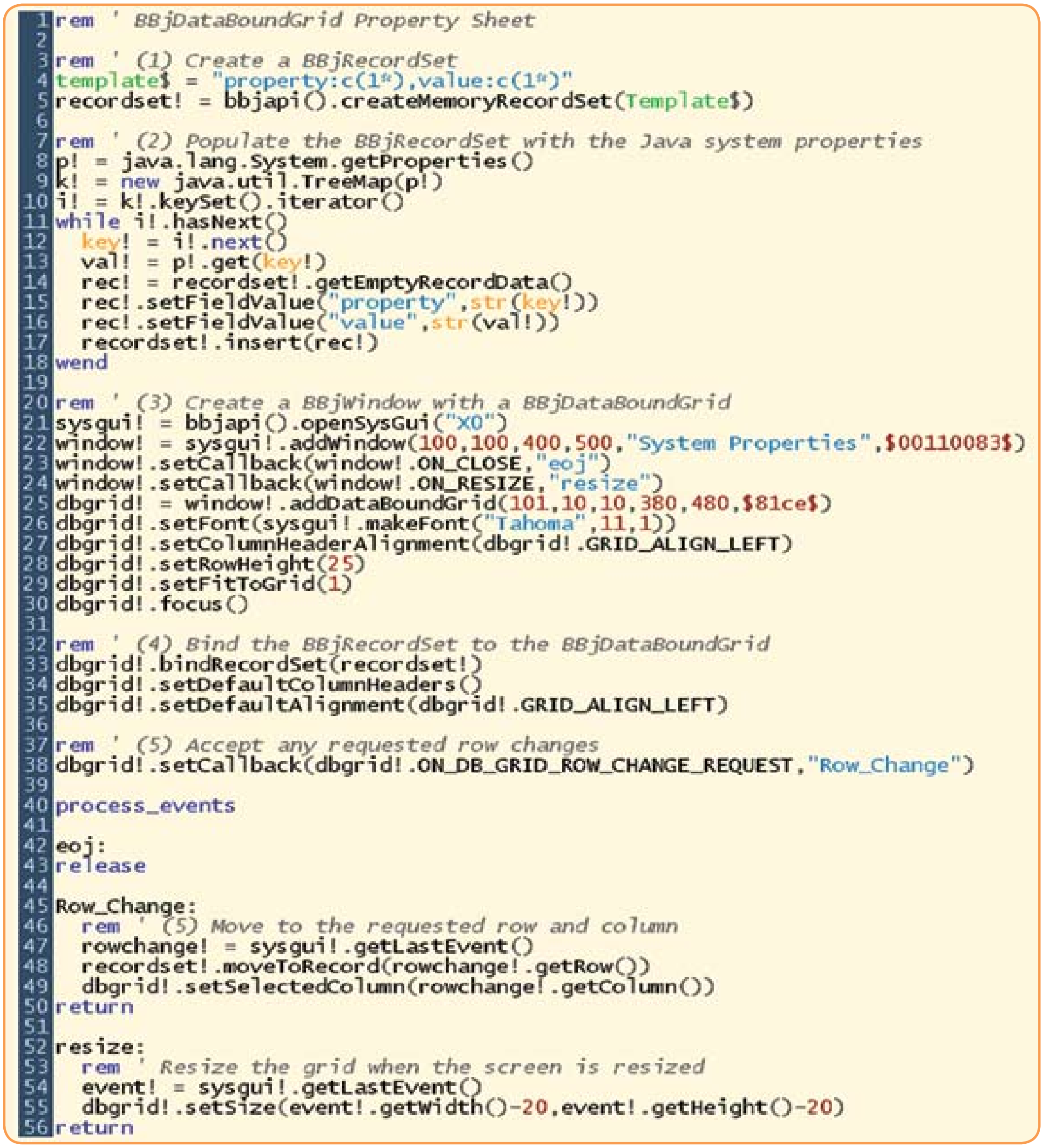
`(2) Populate the BBjRecordSet with the Java system properties`
Lists the current Java system properties in alphabetical order.

`(3) Create a BBjWindow with a BBjDataBoundGrid`

`(4) Bind the BBjRecordSet to the BBjDataBoundGrid`
Binds (associates) the databound grid to the recordset.

`(5) Accept any requested row changes`
Fires a BBjDBGridRowChangeRequestEvent when the user attempts to move to a new row. This sample just displays the data without needing to allow for updates and moves to the requested row and column.

continued...



```
rem ' BBjDataBoundGrid Property Sheet

rem ' (1) Create a BBjRecordSet
template$ = "property:c(1*),value:c(1*)"
recordset! = bbjapi().createMemoryRecordSet(Template$)

rem ' (2) Populate the BBjRecordSet with the Java system properties
p! = java.lang.System.getProperties()
k! = new java.util.TreeMap(p!)
i! = k!.keySet().iterator()
while i!.hasNext()
  key! = i!.next()
  val! = p!.get(key!)
  rec! = recordset!.getEmptyRecordData()
  rec!.setFieldValue("property",str(key!))
  rec!.setFieldValue("value",str(val!))
  recordset!.insert(rec!)
wend

rem ' (3) Create a BBjWindow with a BBjDataBoundGrid
sysgui! = bbjapi().openSysGui("X0")
window! = sysgui!.addWindow(100,100,400,500,"System Properties",$00110083$)
window!.setCallback(window!.ON_CLOSE,"eoj")
window!.setCallback(window!.ON_RESIZE,"resize")
dbgrid! = window!.addDataBoundGrid(101,10,10,380,480,$81ce$)
dbgrid!.setFont(sysgui!.makeFont("Tahoma",11,1))
dbgrid!.setColumnHeaderAlignment(dbgrid!.GRID_ALIGN_LEFT)
dbgrid!.setRowHeight(25)
dbgrid!.setFitToGrid(1)
dbgrid!.focus()

rem ' (4) Bind the BBjRecordSet to the BBjDataBoundGrid
dbgrid!.bindRecordSet(recordset!)
dbgrid!.setDefaultColumnHeaders()
dbgrid!.setDefaultAlignment(dbgrid!.GRID_ALIGN_LEFT)

rem ' (5) Accept any requested row changes
dbgrid!.setCallback(dbgrid!.ON_DB_GRID_ROW_CHANGE_REQUEST,"Row_Change")

process_events

eoj:
release

Row_Change:
  rem ' (5) Move to the requested row and column
  rowchange! = sysgui!.getLastEvent()
  recordset!.moveToRecord(rowchange!.getRow())
  dbgrid!.setSelectedColumn(rowchange!.getColumn())
return

resize:
  rem ' Resize the grid when the screen is resized
  event! = sysgui!.getLastEvent()
  dbgrid!.setSize(event!.getWidth()-20,event!.getHeight()-20)
return
```

Figure 4. BBjDataBoundGrid property sheet – listing (`Properties.src`)

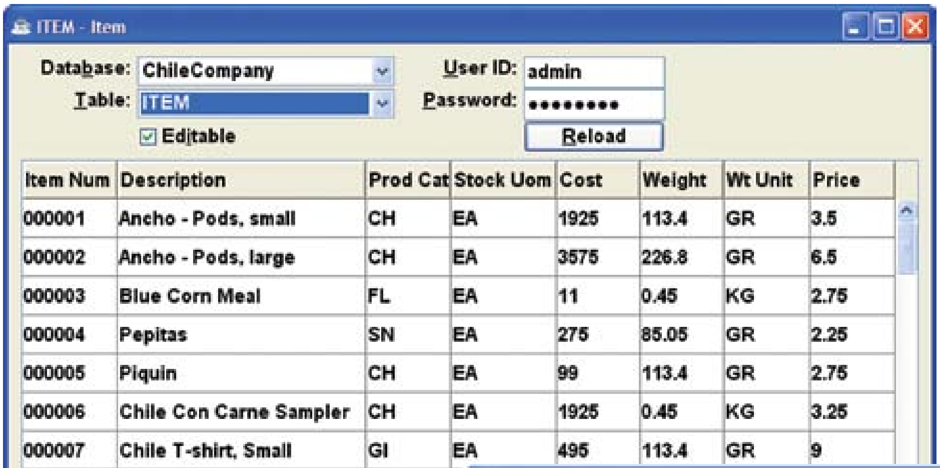
Database Table Editor Sample

The final sample program uses a databound grid to implement a powerful and flexible database editor as displayed in **Figure 5** and **Figure 6**. In addition to being able to change individual fields by editing grid cells, the user can also add, change, or delete records using a dynamically created full-screen editor.

Databound vs. Data-Aware

The databound grid is more powerful and flexible than the older data-aware grid. The data-aware grid automatically writes any changes to the underlying database, without possible programmer intervention. This can allow for slightly less programming, but at the cost of flexibility. With the databound grid, developers always choose when to update changes to the database, allowing more control over data integrity. Furthermore, they can use the databound grid in coordination with other databound controls, all interacting with a single recordset.

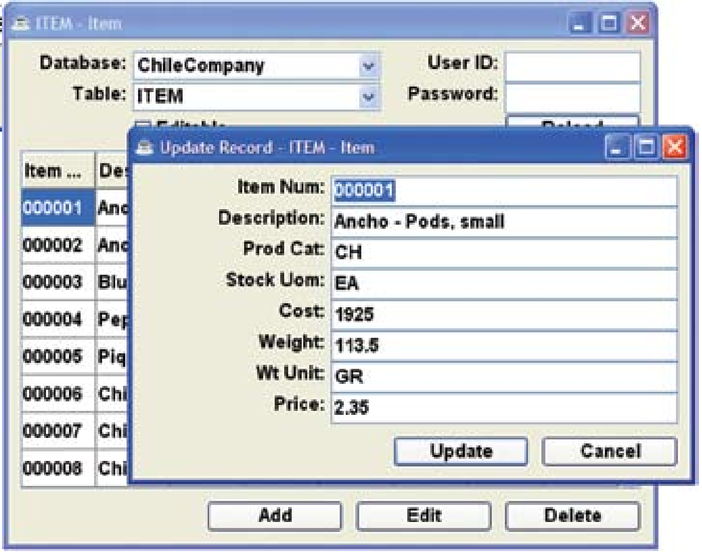
continued...



Database: ChileCompany User ID: admin
Table: ITEM Password: ••••••••
☑ Editable Reload

Item Num	Description	Prod Cat	Stock Uom	Cost	Weight	Wt Unit	Price
000001	Ancho - Pods, small	CH	EA	1925	113.4	GR	3.5
000002	Ancho - Pods, large	CH	EA	3575	226.8	GR	6.5
000003	Blue Corn Meal	FL	EA	11	0.45	KG	2.75
000004	Pepitas	SN	EA	275	85.05	GR	2.25
000005	Piquin	CH	EA	99	113.4	GR	2.75
000006	Chile Con Carne Sampler	CH	EA	1925	0.45	KG	3.25
000007	Chile T-shirt, Small	GI	EA	495	113.4	GR	9
000008	Chile T-shirt, Large	GI	E				

Figure 5. Database table editor (`DBGrid.src`)



Database: ChileCompany User ID:
Table: ITEM Password:

Item ...	De
000001	Anc
000002	Anc
000003	Blu
000004	Pep
000005	Piq
000006	Chi
000007	Chi
000008	Chi

Item Num: 000001
Description: Ancho - Pods, small
Prod Cat: CH
Stock Uom: EA
Cost: 1925
Weight: 113.5
Wt Unit: GR
Price: 2.35

Figure 6. Database table editor – full-screen record editor

Summary

When using databound controls, BBj automatically manages the housekeeping details of copying data from the database record to the screen, and back again. This leaves developers free to focus on the business rules, reducing the time it takes to build powerful and flexible data-oriented applications.

Download the code samples for this article from www.basis.com/advantage/mag-v12n1/databoundgrid.zip

Refer to the online documentation at www.basis.com/onlinedocs/documentation/flash
and search the index for:

BBjRecordSet
BBjDataBoundGrid

Recordsets provide a consistent interface to the data in a BBj multikeyed file, SQL database, or (as of BBj 7.0) an in-memory data structure. For more information about the recordsets, see:

Using the BBjRecordSet
www.basis.com/advantage/mag-v7n3/bbjrecordset.pdf

Why Use the BBjRecordSet?
www.basis.com/advantage/mag-v8n1/recordset.html

Databound controls are BBj GUI controls (e.g. ListButton, InputE, InputN) with an ongoing link to a field in a recordset. For more information about databound controls, review:

BBjDataBound Controls
www.basis.com/advantage/mag-v7n3/bbjdataboundcontrols.pdf

Watch the Form Gen Wizard Trans"form" Data
www.basis.com/advantage/mag-v11n1/FormGenWizard01adv07_Links.pdf

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

Goodbye Event Loop, Hello Process Events

By Jon Bradley

For years, BASIS has provided tools to facilitate writing GUI applications. BASIS is at it again, improving the tools developers use to create their applications!

Background

When BASIS incorporated the functionality of GUIBuilder into the IDE, they decided to continue to read and write the same GBF and ARC files. This makes it easy to move from GUIBuilder to the new, more feature-rich AppBuilder. AppBuilder also uses the same back-end code generator, gb_func, to generate the same READ RECORD-based applications as GUIBuilder. In the preview releases of BBj® 9.0, AppBuilder exposes the generator interface to enable developers to easily modify – or even completely replace – the generator module.

Goodbye, gb_func

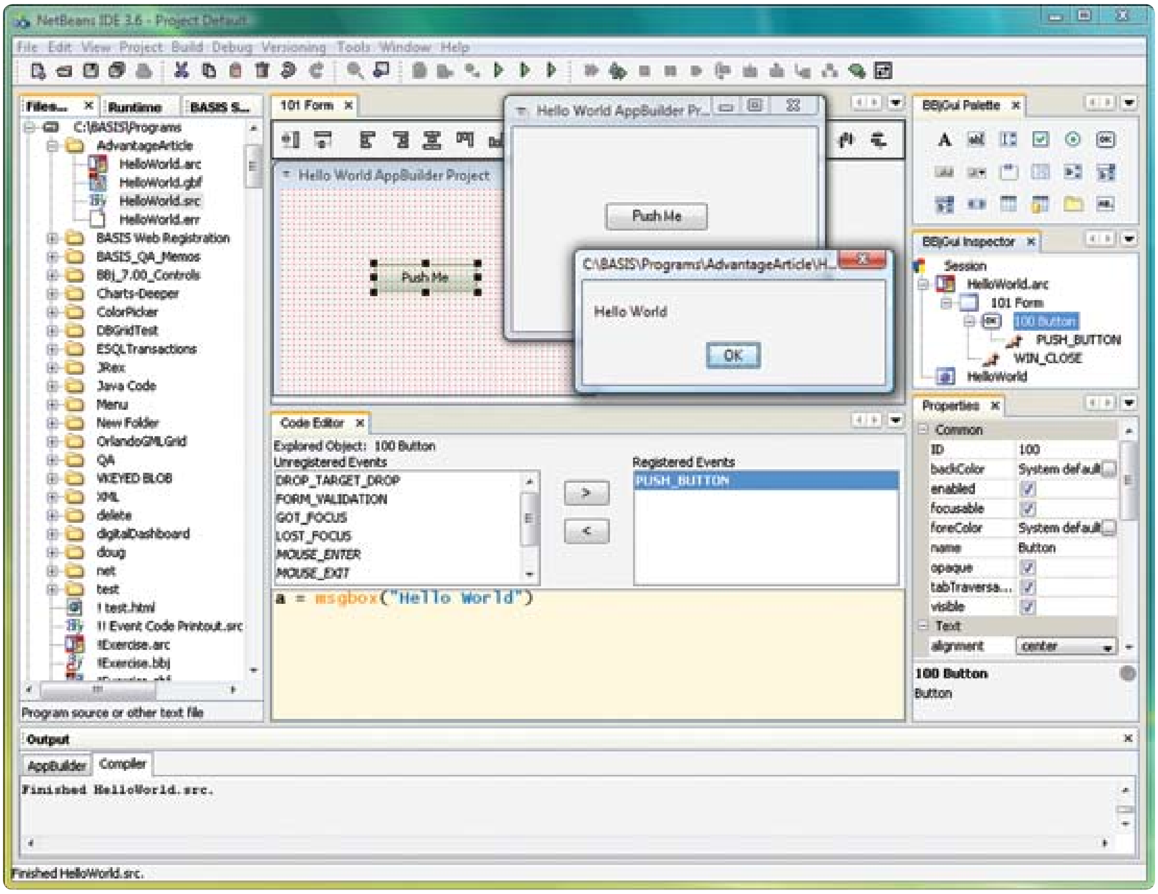
Both GUIBuilder and AppBuilder use a generator program to convert the .gbf and .arc files into a deliverable application. Prior to BBj 9.0, AppBuilder always used the same gb_func generator program as GUIBuilder, thus it produced the exact same application code. BBj has so many new language constructs ranging from PROCESS_EVENTS to custom objects that BASIS wanted to create a new generator program to exploit the new language features and generate more powerful, readable, and efficient programs.

Hello, ProcessEventsBuilder

Instead of dictating what form the newly generated application code takes, BASIS makes it easy to write a custom generator program while supplying two sample generator programs to get started. The ProcessEventsBuilder generator and its close cousin, LegacyPEBuilder, both use the PROCESS_EVENTS method for dispatching events. This method represents a more modern, succinct, and efficient way to handle events.

When using PROCESS_EVENTS, developers register callbacks for the specific events on specific controls for which they are interested. The dispatch of that event to a given routine happens automatically, reducing the amount of boilerplate

continued...



Hello World

Explored Object: 100 Button

Unregistered Events
DROP_TARGET_DROP
FORM_VALIDATION
GOT_FOCUS
LOST_FOCUS
MOUSE_ENTER
MOUSE_EXIT

Registered Events
PUSH_BUTTON

```
a = msgbox("Hello World")
```

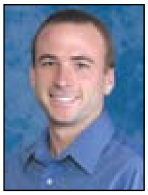
Common	
ID	100
backColor	System default
enabled	☑
focusable	☑
foreColor	System default
name	Button
opaque	☑
tabTraversa...	☑
visible	☑
Text	
alignment	center

100 Button
Button

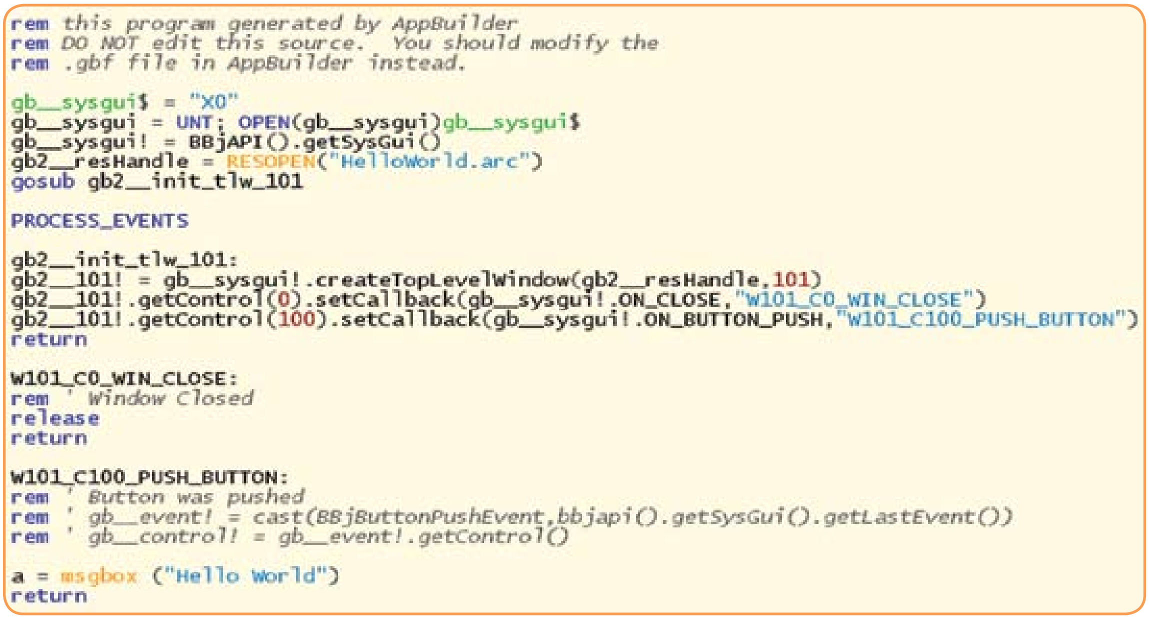
```
Finished HelloWorld.src.
```

Finished HelloWorld.src.

Figure 1. "Hello World" AppBuilder project



Jon Bradley
Software Engineer



```
rem this program generated by AppBuilder
rem DO NOT edit this source.  You should modify the
rem .gbf file in AppBuilder instead.

gb__sysgui$ = "X0"
gb__sysgui = UNT; OPEN(gb__sysgui)gb__sysgui$
gb__sysgui! = BBjAPI().getSysGui()
gb2__resHandle = RESOPEN("HelloWorld.arc")
gosub gb2__init_tlw_101

PROCESS_EVENTS

gb2__init_tlw_101:
gb2__101! = gb__sysgui!.createTopLevelWindow(gb2__resHandle,101)
gb2__101!.getControl(0).setCallback(gb__sysgui!.ON_CLOSE,"W101_C0_WIN_CLOSE")
gb2__101!.getControl(100).setCallback(gb__sysgui!.ON_BUTTON_PUSH,"W101_C100_PUSH_BUTTON")
return

W101_C0_WIN_CLOSE:
rem ' Window Closed
release
return

W101_C100_PUSH_BUTTON:
rem ' Button was pushed
rem ' gb__event! = cast(BBjButtonPushEvent,bbjapi().getSysGui().getLastEvent())
rem ' gb__control! = gb__event!.getControl()

a = msgbox ("Hello World")
return
```

Figure 2. Resulting code produced from building the AppBuilder project

code and the number of instructions required to process an event in the interpreter. The GUI sends fewer messages to the server, reducing network utilization and improving responsiveness.

The ProcessEventsBuilder generator program produces object-oriented code, resulting in a program that is easier to read and debug. It makes use of BBjEvent objects, which provide intuitive method names for exposing the information – information that previously appeared in hard-to-understand notify string flags.

Case in Point

In the simple "Hello World" AppBuilder project shown in **Figure 1**, the `ProcessEventsBuilder.src` generator program produced 24 lines of code shown in **Figure 2**.

This increases the readability and eases the debugging process because the resultant program is much smaller. The gb_func-generated program for the same AppBuilder project is a whopping 712 lines of code! This enormous difference is due to the large amount of code gb_func-generated programs needed in Visual PRO/5® to get and set information for the SYSGUI system. Generating a PROCESS_EVENTS program also allows the control validation system to work normally, without having to add the extra call to setCallback in the init block, which is necessary when creating a READ RECORD-based program.

Building a Custom Generator Program

BASIS provides the new AdvancedGBFWrapper Java object to expose the contents of the GBF file to the generator program in an object-oriented way. This makes writing new generator programs relatively painless. For instance, the ProcessEventsBuilder generator program is only 200 lines of code. The BASIS-provided generator programs use this same model and present a good springboard for those developers looking to take the plunge and create their own generator program.

By default, BBj continues to use the gb_func program generator. Select a different generator program via **Tools > Options > Form & AppBuilder Settings > Custom Generator Program** in the BASIS IDE.

Summary

Because the generator programs are customizable, it is easy to leverage the BBj language enhancements to create more readable and powerful applications. The application will benefit from a substantial reduction in the lines of code generated; in our "Hello World" example, the number of lines went down from 712 to just 24! Some developers have applied their own enhancements and modifications to gb_func to address their needs to create custom application code. The **AdvancedGBFWrapper** meets this need by facilitating the creation of a custom generator program that produces the application code. This delivers the desired structure when using AppBuilder for WYSIWYG editing of applications within the BASIS IDE.

Read more about design guidelines of a generator in *Working with Custom Program Generators*
www.basis.com/solutions/UsingCustomAppBuilderGeneratorPrograms.pdf

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

Take a Plunge Into SQL-Transactions

By Christopher Hardekopf

Transactions in SQL are a useful tool for preserving data integrity. With BASIS' ESQL tables, BBj® developers are now able to exploit that tool without the need to use a third party database. To use transactions appropriately, it is necessary to understand the important concepts in this article regarding their benefits and limitations.

Transactions Reviewed

Simply stated, transactions are a mechanism for grouping database operations into a single unit. This might be as straightforward as a single update or as complicated as several selects, inserts, and updates on different ESQL tables. This group of operations can then be either committed or rolled back. Committing a transaction makes every operation persistent, while rolling the transaction back will reverse every operation. Either will unlock all resources locked by the transaction. If the server machine crashes while a transaction is still in progress, the transaction operations automatically roll back, leaving the ESQL tables in a known good state. This means that either all of the operations in a transaction will persist in the database or none of the operations will take effect. The developer never has to worry that only part of a list of operations will persist in the database.

In addition, developers can create savepoints – intermediate stages of a transaction to which they can roll back – leaving the transaction as a whole open rather than rolling back the entire transaction. Later, they can choose to either commit or roll back all of the other operations in the transaction. Transactions are in auto-commit mode unless the developer chooses to use transactions explicitly. This simply means that each individual SQL statement runs inside its own transaction, generally transparent, that is automatically committed when the statement completes.

Things to Keep in Mind

Using transactions requires ESQL tables. BBj transactions need substantial file support to work properly so they do not work with traditional BBx® files. BBj transactions are table-oriented and do not handle SQL meta operations. Developers cannot roll back operations such as create database, drop database, create table, and drop table in a transaction.

Because of the data integrity guarantees provided by transactions, they must lock various database resources as the data is read and changed. After all, two transactions cannot change the same data since the client might choose to roll back either transaction. This would lead to a lot of confusion and unpredictable behavior. As a result, individual transactions should only be maintained for short periods of time. In particular, never leave a transaction open while an application waits for user input. While this is possible, it leads to slowness as it blocks several users waiting on another user. It is also valuable for clients to use read-only database connections when they do not need to write data to the database. Transaction locking for a read-only connection is much less restrictive than locking in a read-write connection. Read-only database connections allow clients to share database resources much more efficiently.

Transaction Isolation

The primary configuration for individual transactions is the transaction isolation level, which controls accessibility of the data that other transactions are concurrently modifying. In other words, transactions view data based on their own isolation levels rather than the isolation levels of other transactions. The Java API provides five different transaction isolation levels, each of which has different blocking behavior in accordance with the transactions' visibility. In general, the more restrictive the reads, the more blocking is required. The transaction levels range from TRANSACTION_NONE, which indicates that transactions are not used or are unsupported, to TRANSACTION_SERIALIZABLE, which is the strictest.

Summary

If you have not dipped your toes into ESQL files, now is definitely the time to dive in, head first. With the latest addition of transactional support, ESQL files further the concept of data availability and reliability. Because the JDBC API and many third party tools support SQL transactions, you can take the big plunge and start working with transactions today!

For more about ESQL tables and isolation levels, refer to the online documentation at www.basis.com/onlinedocs/documentation/flash and search the index for:

ESQL Overview
Connecting Using the JDBC Driver

Read the *BASIS International Advantage* article
Implementing Journaled Files and Transaction Tracking
www.basis.com/advantage/mag-v6n1/journaled.html

Chris Hardekopf
Software Engineer

BARISTA®

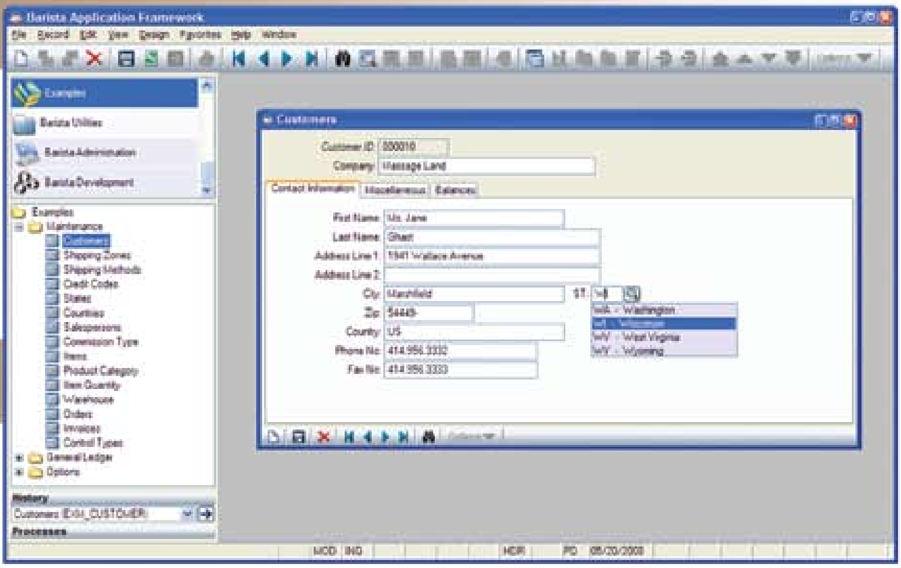
GUI Application Framework

Brews

- Fresh new GUI applications
- GUI-transformed CUI applications
- Modernized existing GUI applications
- Cross-platform GUI applications

Benefits

- Radically reduces lines of code
- Saves time
  - Creates working entry forms in minutes
  - Re-uses existing update and report programs
- Easily links tables and forms
- Ensures visual consistency and quality of applications
- Creates global applications with the multilanguage feature
- Enables broad document output functionality such as e-mail/fax/PDF

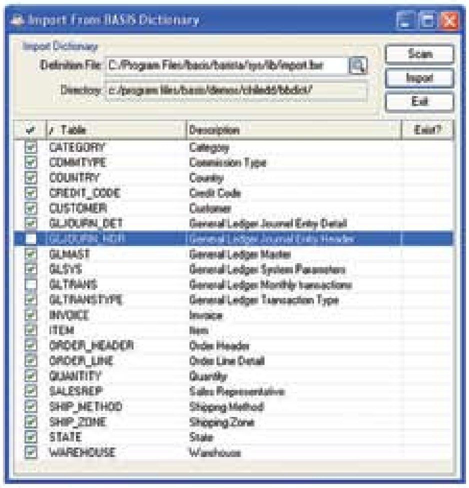


History: Customers (EXM_CUSTOMER)

Customer ID: 000010
Company: Massage Land
First Name: Ms. Jane
Last Name: Ghast
Address Line 1: 1941 Wallace Avenue
Address Line 2:
City: Marshfield
ST: WI
Zip: 54449
Country: US
Phone No: 414.956.3332
Fax No: 414.956.3333

WA - Washington
WI - Wisconsin
WV - West Virginia
WY - Wyoming

Customer Data Maintenance showing field auto-completion capability

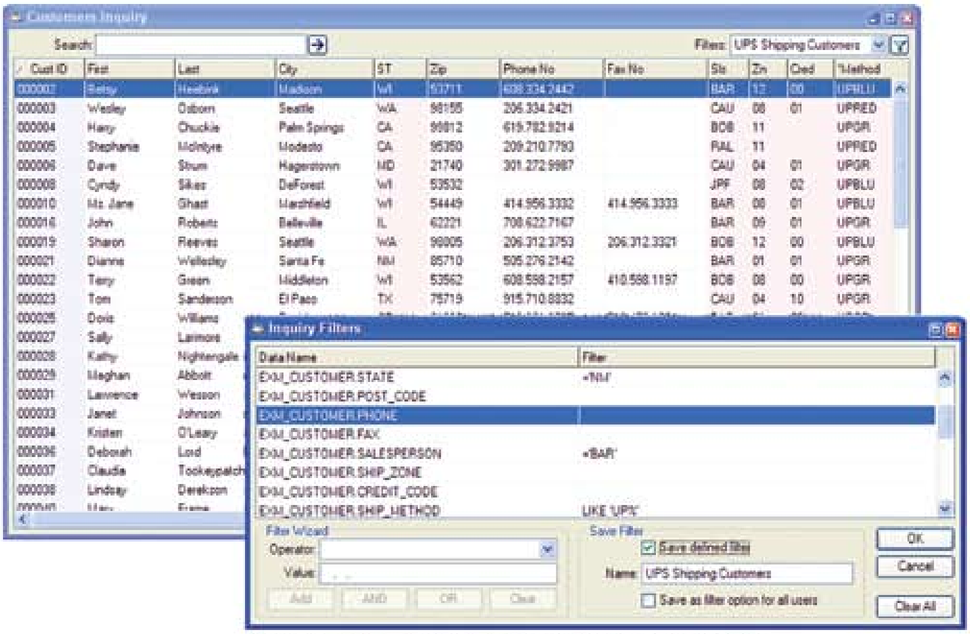


Definition File: C:/Program Files/basis/barista/sys/lib/import.bar
Directory: c:/program files/basis/demos/chiledd/bbdict/

✓	Table	Description	Exist?
☑	CATEGORY	Category	
☑	COMMTYPE	Commission Type	
☑	COUNTRY	Country	
☑	CREDIT_CODE	Credit Code	
☑	CUSTOMER	Customer	
☑	GLJOURN_DET	General Ledger Journal Entry Detail	
□	GLJOURN_HDR	General Ledger Journal Entry Header	
☑	GLMAST	General Ledger Master	
☑	GLSYS	General Ledger System Parameters	
□	GLTRANS	General Ledger Monthly transactions	
☑	GLTRANSTYPE	General Ledger Transaction Type	
☑	INVOICE	Invoice	
☑	ITEM	Item	
☑	ORDER_HEADER	Order Header	
☑	ORDER_LINE	Order Line Detail	
☑	QUANTITY	Quantity	
☑	SALESREP	Sales Representative	
☑	SHIP_METHOD	Shipping Method	
☑	SHIP_ZONE	Shipping Zone	
☑	STATE	State	
☑	WAREHOUSE	Warehouse	

Import tables from BASIS Data Dictionary

> *"We were very pleased with the Barista presentation and training in Costa Rica. Barista is a very stable tool that allows coding with great simplicity and speed. Those developers that select not to use Barista will miss out on greater programming productivity."*
>
> *- Jorge Garro, Owner*
> *Unisoft*



Search:
Filters: UPS Shipping Customers

Cust ID	First	Last	City	ST	Zip	Phone No	Fax No	Sls	Zn	Cred	Method
000002	Betsy	Heebink	Madison	WI	53711	608.334.2442		BAR	12	00	UPBLU
000003	Wesley	Osborn	Seattle	WA	98155	206.334.2421		CAU	08	01	UPRED
000004	Harry	Chuckie	Palm Springs	CA	95812	619.782.9214		BOB	11		UPGR
000005	Stephanie	McIntyre	Modesto	CA	95350	209.210.7793		RAL	11		UPRED
000006	Dave	Strum	Hagerstown	MD	21740	301.272.9987		CAU	04	01	UPGR
000008	Cyndy	Sikes	DeForest	WI	53532			JPF	08	02	UPBLU
000010	Ms. Jane	Ghast	Marshfield	WI	54449	414.956.3332	414.956.3333	BAR	08	01	UPBLU
000016	John	Roberts	Belleville	IL	62221	708.622.7167		BAR	09	01	UPGR
000019	Sharon	Reeves	Seattle	WA	98005	206.312.3753	206.312.3321	BOB	12	00	UPBLU
000021	Dianne	Wellesley	Santa Fe	NM	85710	505.276.2142		BAR	01	01	UPGR
000022	Terry	Green	Middleton	WI	53562	608.558.2157	410.558.1157	BOB	08	00	UPGR
000023	Tom	Sanderson	El Paso	TX	75719	915.710.8832		CAU	04	10	UPGR
000025	Doris	Williams									
000027	Sally	Larmore									
000028	Kathy	Nightengale									
000029	Maghan	Abbott									
000031	Lawrence	Wesson									
000033	Janet	Johnson									
000034	Kristen	O'Leary									
000036	Deborah	Lord									
000037	Claudia	Tookeypatch									
000038	Lindsay	Derekson									

Data Name	Filter
EXM_CUSTOMER.STATE	='NM'
EXM_CUSTOMER.POST_CODE	
EXM_CUSTOMER.PHONE	
EXM_CUSTOMER.FAX	
EXM_CUSTOMER.SALESPERSON	='BAR'
EXM_CUSTOMER.SHIP_ZONE	
EXM_CUSTOMER.CREDIT_CODE	
EXM_CUSTOMER.SHIP_METHOD	LIKE 'UPS'

Operator:
Value:
☑ Save defined filter
Name: UPS Shipping Customers
□ Save as filter option for all users

Automatic SQL-based Inquiry System with filtering capability

T Minus 0 – BASIS LaunchDock Blasts Off

By Nick Decker

Each time another BASIS conference rolls around, we start wracking our brains to come up with a new way of organizing the numerous demonstration programs into the workbench application. One essential design consideration is that the workbench program should incorporate some of the latest BBj® features, which is why some of the past workbench applications took advantage of constructs like MDI (multiple document interface), expandable palettes, versatile toolbars, and more. Given that BBj now supports Java GUI objects on the client's machine, we decided to make BBj's new ClientObject capability the centerpiece of the workbench application.

ClientObjects Break New Ground

In addition to a radically different interface, we wanted to include plenty of 'eye candy' to the workbench program. After all, form is often just as important as function especially when you are showing off. Just ask all those Olympic gymnastic judges.

In order to accomplish these goals we did something that has never been possible in BBj up until now – we created the top level window as a semi-transparent, alpha-blended window replete with a drop shadow, rendered based upon the contents of a graphics file (**Figure 1**).

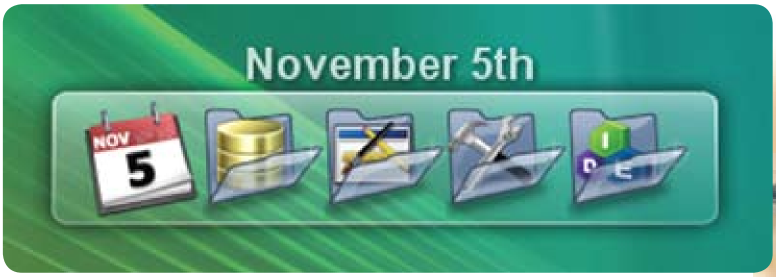


Figure 1. The BASIS LaunchDock menus for the November 5th session of TechCon2007

To decrease our development time dramatically, we took advantage of third party freeware Java libraries that use native code to create a transparent JWindow. This is significant for a couple of reasons. First, the resultant window is a BBj ClientObject, meaning that it is a Java-based GUI component that exists in the Client's JVM and displays on the Client's machine. Second, since the window isn't a BBjWindow and it exists in a completely different JVM than the BBj Interpreter is running in (which is on the server machine), we have to do a little extra work to connect the Java window to the BBj program so that we can interact with it successfully.

A Closer Look at the Technologies Involved

The LaunchDock is comprised of several nested constructs and technologies. At the core of the LaunchDock is a Java program that handles all of the user interaction. This includes the GUI components, such as the Java-based transparent window, as well as the events - including mouse clicks and drags. One of the first things the Java program does is create the transparent window. To do this, it instantiates a TransparentWindow object. The TransparentWindow object extends a typical Java JWindow, but employs code from the Java Native Access (JNA) libraries to make the resultant window transparent. The transparent window serves as a blank canvas on which the Java program paints the background image and icons. Because the window is transparent, you can see other windows and the wallpaper underneath the window.

Nick Decker
Engineering
Supervisor

The LaunchDock Java program takes advantage of other free Java libraries such as the SwingX collection of components. The TimingFramework library facilitates the timing and animations that the LaunchDock uses for icon hover effects. In addition to adding some pizzazz to the LaunchDock, the animations enhance the user's experience. They provide visual feedback to the user in response to events such as zooming in on an icon when the mouse hovers over it or bouncing an icon after the user clicks on it. The SwingX libraries also enable the LaunchDock to provide additional effects for the user-supplied icons. This means that the LaunchDock can take a stock icon and enhance it on the fly by adding a drop shadow or a reflection as shown in **Figure 2**.

continued...

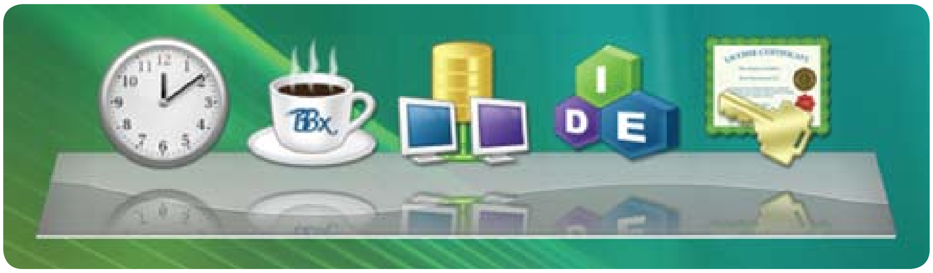


Figure 2. The LaunchDock displaying real-time reflections of the icons

Passing User-Generated Events Back to the BBj Program

Because the LaunchDock's window is a Java-based JWindow instead of a BBjWindow, we have to create a way for LaunchDock to notify the BBj program that instantiates it whenever the user interacts with the dock. If we had used a standard BBjWindow, we could easily register for callback events on the window to find out when the user clicked on it or one of the icons. But in this case, the LaunchDock's window is a JWindow that exists in a different JVM on the client's machine, so the LaunchDock utilizes Java's Listener mechanism to accomplish this communications link. The LaunchDock Java program exposes methods that allow other objects to add themselves to a collection of listeners, and then relays important user-generated events such as mouse clicks to those registered objects. The result is that the BBj program receives custom events from the LaunchDock and acts upon those events in the appropriate manner, whether it is launching the desired program or showing the About Box.

Customizing the LaunchDock

While we primarily programmed the LaunchDock to serve as the TechCon2007 workbench application, we knew that if we designed it in a flexible and extensible fashion, it could be a powerful tool, extending its usefulness well beyond the conference. With that in mind, we came up with the concept of an Archive object to define the contents of the LaunchDock. In simple terms, an Archive is a hierarchical collection of folders, programs, and icons that describes all of the programs that the LaunchDock should display. This makes it easy to use the LaunchDock for several applications - just pass it one Archive for the BASIS demos and another to launch a completely different set of programs. Users can even create and manage their own Archives with our custom Archive editor. We have exploited this capability at BASIS and use the LaunchDock for our in-house program menu. BASIS employees are also free to customize their copy of the Java Web start deployed LaunchDock by adding their Windows, Linux, or Apple Mac office productivity application programs to their individual copy of the Archive.

In addition to using the LaunchDock for various purposes, programmers can customize its appearance via several built-in methods. Dozens of aspects are configurable such as the graphics used for the background as well as the sizing and placement of the icons. Developers are able to tweak each parameter to alter the LaunchDock's appearance and behavior.

Users have their preferences as well, so the LaunchDock's popup menus enable the end user to set the location and orientation of the LaunchDock. Users can then decide what works best with their desktop – whether they have the LaunchDock docked to the top, bottom, left, or right sides of the screen as shown in **Figure 3**. If that still does not fit their needs, they have the option of undocking it and moving it anywhere on the desktop by dragging it with their mouse. Moreover, if the dock needs to be accessible all of the time, selecting the On Top option prevents other windows from obscuring it, thus causing it to be visible all of the time.

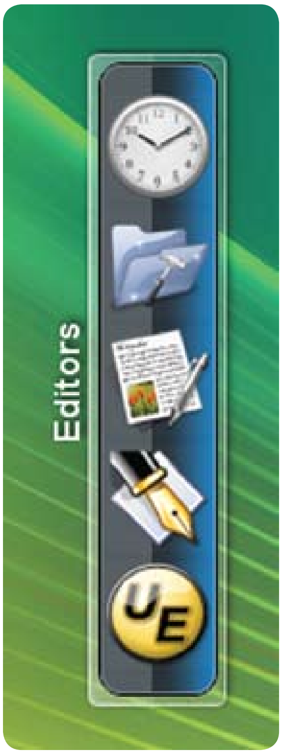


Figure 3. The LaunchDock in a vertical orientation

Summary

Without regard to the technology employed, the primary goal for any conference workbench application is to come up with an intuitive way to arrange and separate the demo programs and still make it easy to navigate to a particular demo. The BASIS Demo LaunchDock takes advantage of BBj's new ClientObject capability to do exactly that – in a single compact user interface. Do not let the simplistic nature of the interface fool you though, as there are countless advanced technologies utilized under-the-hood to make this the most sophisticated workbench program ever.

To see the LaunchDock in action, download the demos with BBj from
www.basis.com/products/bbj/download.html and select "Demos" in the Optional File section
After completing the installation, select **BBj > Demos > LaunchDock** from the BASIS folder
to run the demo.

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

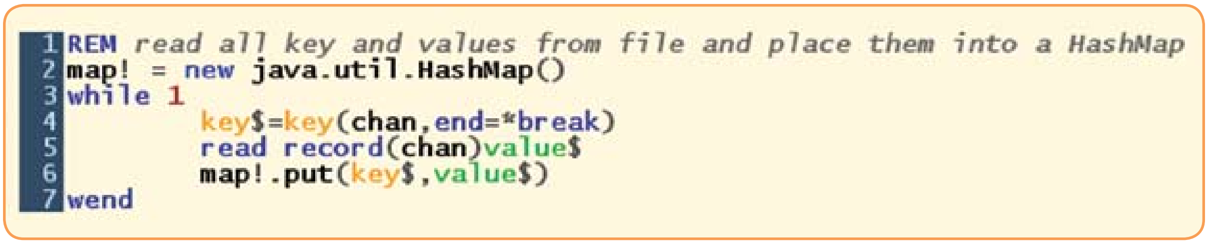
Blowing the Doors Wide Open With ClientObjects

By David Wallwork

BBj® introduces enhancements and features with each new release. One of these features, embedded Java code, allows the developer to embed Java code within their BBj programs in order to take advantage of Java's advanced data types, JVM-specific information, and the huge body of readily-available Java code and libraries. In BBj 8.0, the concept of embedding Java code is extended through the introduction of ClientObjects. This article explores how developers can use the new ClientObject feature to create and implement client-side Java Objects including third party classes as well as custom classes; GUI controls as well as non-GUI Objects.

The Need for Client Objects

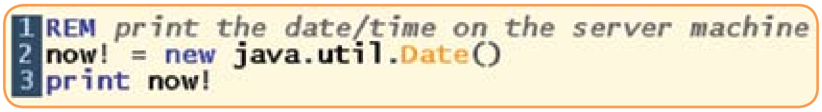
Consider a sample program (see **Figure 1**) that reads key-value pairs from a channel and places them into a Java HashMap.



```
1 REM read all key and values from file and place them into a HashMap
2 map! = new java.util.HashMap()
3 while 1
4         key$=key(chan,end=*break)
5         read record(chan)value$
6         map!.put(key$,value$)
7 wend
```

Figure 1. Read and place key-value pairs into a Java HashMap

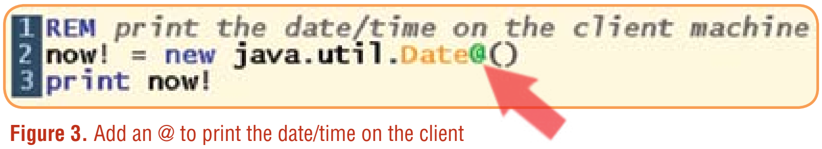
This example creates a HashMap within the JVM of the interpreter, which is often what developers want. However, there are times when they need to create a Java Object within the JVM of the client rather than within the JVM of the server. For instance, to print the current time, one might write the code shown in **Figure 2**.



```
1 REM print the date/time on the server machine
2 now! = new java.util.Date()
3 print now!
```

Figure 2. Print the current date and time for the server, rather than the client

This is all well and good if the server and the client are in the same time zone, but what happens when the server and the client are in different time zones? BBj 8.0's ClientObjects come to the rescue. To print the date and time of the client, the program must use ClientObjects. ClientObjects use an @ symbol to indicate that a Java Object should be created within the *client* JVM rather than within the *server* JVM. Simply adding the @ symbol to our example in **Figure 3** creates a Date Object on the client machine and prints the date and time of the client.



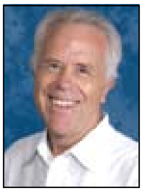
```
1 REM print the date/time on the client machine
2 now! = new java.util.Date@()
3 print now!
```

Figure 3. Add an @ to print the date/time on the client

Background and Terminology

Because the interaction between the BBj server and the BBj client is seamless, most developers do not think about the difference between the server-side environment and the client-side environment. In order to understand ClientObjects, it is important to understand this distinction.

In most common deployments, the client and the server are on different machines, so it is easy to distinguish between them. Even when the client and server are on the same machine; even if they are running within the same JVM, we still make a distinction. We refer to any object created within the JVM of BBjServices as a server-side object. Any object created within the JVM of the client is a client-side object. For clarity of discussion, assume that the client and server are running on separate machines.



David Wallwork
Senior Software Engineer

Under the covers, ClientObjects are simply Remote Method Invocation (RMI) handles to the client-side Java Object. When executing the code in **Figure 3**, line 2 creates two Java objects; one is a client-side instance of java.util.Date, the other is a ClientObject that is an RMI handle to the client-side instance of java.util.Date.

The ClientObject has the same public methods as the client-side object so the BBj program can call the same methods on `now!` in **Figure 3** as it can call on `now!` in **Figure 2**. Though both variables look the same to the program, the variable in **Figure 2** holds a server-side object while the variable in Example 3 holds an RMI handle to a client-side object.

continued...

This distinction – whether an Object is in the client JVM or in the server JVM – is especially important when using GUI Objects. A GUI Object that is created in the server JVM will display on the server console. A GUI Object that is created in the client JVM will display on the client console.

Seeing Client Objects is Believing

BASIS created several demo programs to illustrate the myriad of features introduced in BBj 8.0. Many of these demos use ClientObjects to perform in ways that were not possible prior to 8.0.

First, run the "LaunchDock" demo† that is based on ClientObjects. Notice the floating tool button that displays menu selections as you mouse over the buttons. Next, launch the "Web Browser" demo† to see a fully-featured Web browser controlled by a BBj program. As with all the demos, the full code is available for your review in its appropriately named folder.

Here are several noteworthy aspects about the "Embedded Web Browser" demo program:

- The actual GUI component seen on the client screen is an instance of `org.jdesktop.jdic.browser.WebBrowser`, which is an open-source Java class downloadable free of charge from www.jdic.dev.java.net.

  Using this class points out a powerful benefit of ClientObjects – developers can now instantly create applications with controls that are not part of the BBj language. Instead of requesting that BASIS write a Web browser control from scratch for BBj, developers can download a Java-based Web browser from the Internet and begin using it right away.

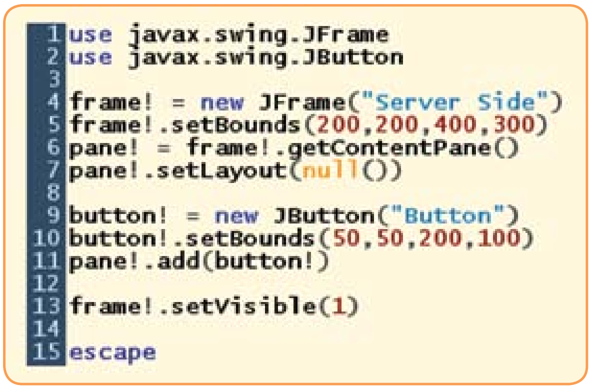
- The BBjWrappedJComponent wraps the WebBrowser object resulting in a BBjControl with a control ID and all the methods defined for a BBjControl

- The BBj program responds to Java events generated by the WebBrowser object

The "Embedded Web Browser" demo† is a fully developed application and at first glance may seem intimidating to the reader who is not familiar with CustomObjects and ClientObjects. However, the following sections provide simpler examples of the basic concepts used in this demo.

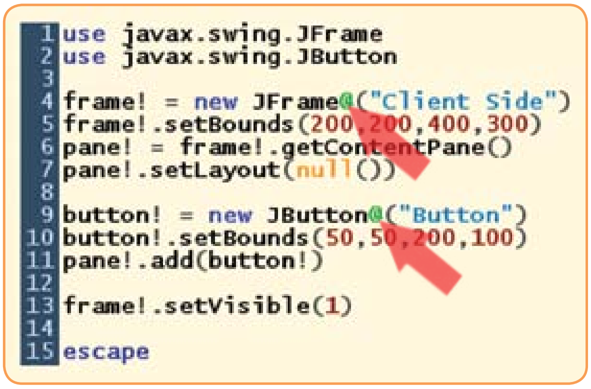
Displaying Java GUI Components on the Client

Since the introduction of embedded Java (server-side objects), a BBj program has been able to display Java GUI components as illustrated in **Figure 4.**



```
use javax.swing.JFrame
use javax.swing.JButton

frame! = new JFrame("Server Side")
frame!.setBounds(200,200,400,300)
pane! = frame!.getContentPane()
pane!.setLayout(null())

button! = new JButton("Button")
button!.setBounds(50,50,200,100)
pane!.add(button!)

frame!.setVisible(1)

escape
```

Figure 4. Display the Java GUI component on the server



```
use javax.swing.JFrame
use javax.swing.JButton

frame! = new JFrame@("Client Side")
frame!.setBounds(200,200,400,300)
pane! = frame!.getContentPane()
pane!.setLayout(null())

button! = new JButton@("Button")
button!.setBounds(50,50,200,100)
pane!.add(button!)

frame!.setVisible(1)

escape
```

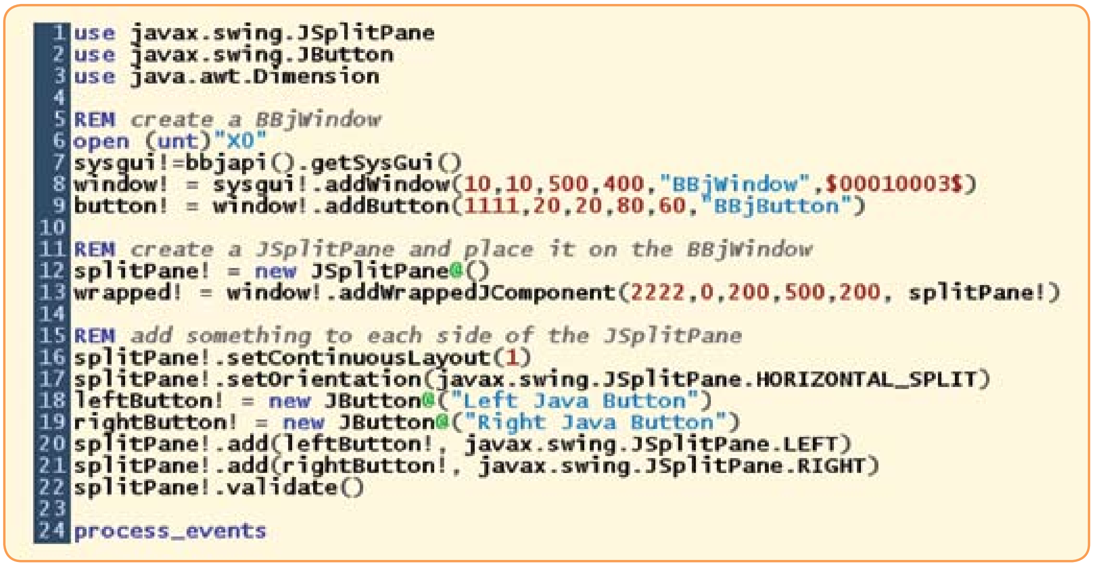
Figure 5. Display the Java GUI component on the client

The problem with the sample in **Figure 4,** however, is that the JFrame and JButton display on the server, which is not of much use when the client is on a different machine. Simply adding an @ symbol as shown in **Figure 5** changes the server-side objects into ClientObjects so that these entities display on the client machine rather than on the server machine.

continued...

Introducing BBjWrappedJComponent and Adding a Java GUI Component to a BBjWindow

BBj introduced a new BBjControl named BBjWrappedJComponent in version 8.0 so developers can ’wrap’ a JComponent and place it onto a BBjWindow. The program in **Figure 6** creates a BBjWindow and adds a BBjButton to the window. It then creates a JSplitPane, wraps the JSplitPane in a BBjWrappedJComponent, and places the component onto the BBj Window.



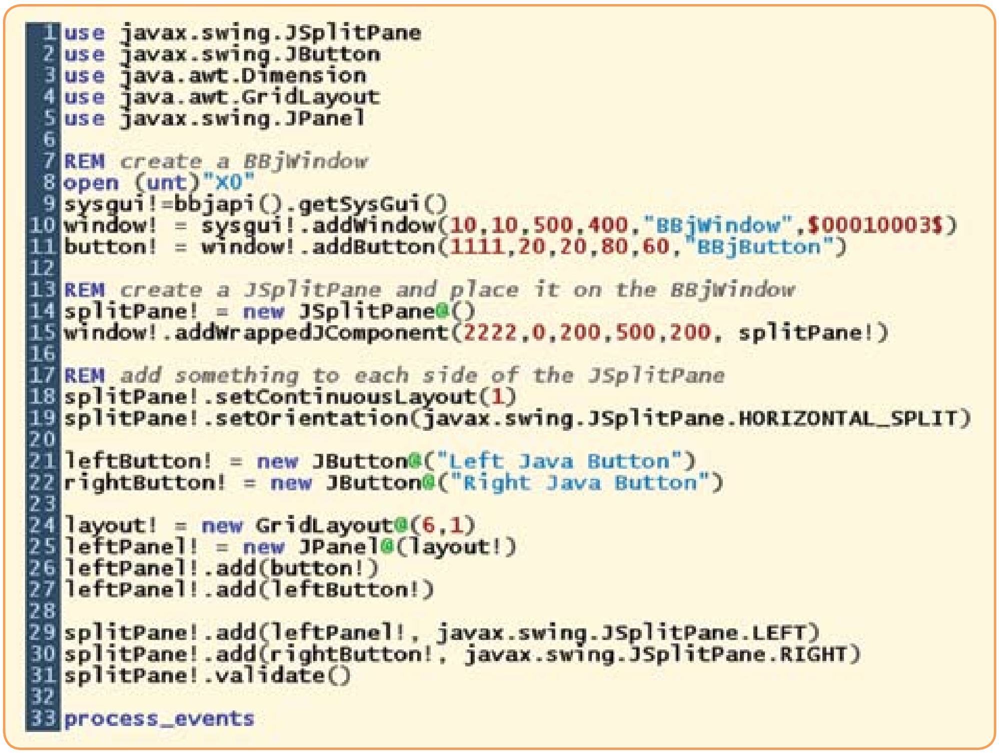
```
 1 use javax.swing.JSplitPane
 2 use javax.swing.JButton
 3 use java.awt.Dimension
 4
 5 REM create a BBjWindow
 6 open (unt)"X0"
 7 sysgui!=bbjapi().getSysGui()
 8 window! = sysgui!.addWindow(10,10,500,400,"BBjWindow",$00010003$)
 9 button! = window!.addButton(1111,20,20,80,60,"BBjButton")
10
11 REM create a JSplitPane and place it on the BBjWindow
12 splitPane! = new JSplitPane@()
13 wrapped! = window!.addWrappedJComponent(2222,0,200,500,200, splitPane!)
14
15 REM add something to each side of the JSplitPane
16 splitPane!.setContinuousLayout(1)
17 splitPane!.setOrientation(javax.swing.JSplitPane.HORIZONTAL_SPLIT)
18 leftButton! = new JButton@("Left Java Button")
19 rightButton! = new JButton@("Right Java Button")
20 splitPane!.add(leftButton!, javax.swing.JSplitPane.LEFT)
21 splitPane!.add(rightButton!, javax.swing.JSplitPane.RIGHT)
22 splitPane!.validate()
23
24 process_events
```

Figure 6. Wrap the JSplitPane in a BBjWrappedJComponent

The `wrapped!` variable on line 13 is a BBj component and has a control ID. A developer can manipulate it using the methods available on any BBj component while still being able to manipulate the ’wrapped’ JSplitPane with all the public methods of a JSplitPane.

Adding a BBjControl to a Java Component

BBj allows a BBjControl to be passed to the add method of a JComponent. The example in **Figure 7** removes the BBjControl from the BBjWindow and places it directly onto a JPanel that appears on the left side of the JSlider.



```
 1 use javax.swing.JSplitPane
 2 use javax.swing.JButton
 3 use java.awt.Dimension
 4 use java.awt.GridLayout
 5 use javax.swing.JPanel
 6
 7 REM create a BBjWindow
 8 open (unt)"X0"
 9 sysgui!=bbjapi().getSysGui()
10 window! = sysgui!.addWindow(10,10,500,400,"BBjWindow",$00010003$)
11 button! = window!.addButton(1111,20,20,80,60,"BBjButton")
12
13 REM create a JSplitPane and place it on the BBjWindow
14 splitPane! = new JSplitPane@()
15 window!.addWrappedJComponent(2222,0,200,500,200, splitPane!)
16
17 REM add something to each side of the JSplitPane
18 splitPane!.setContinuousLayout(1)
19 splitPane!.setOrientation(javax.swing.JSplitPane.HORIZONTAL_SPLIT)
20
21 leftButton! = new JButton@("Left Java Button")
22 rightButton! = new JButton@("Right Java Button")
23
24 layout! = new GridLayout@(6,1)
25 leftPanel! = new JPanel@(layout!)
26 leftPanel!.add(button!)
27 leftPanel!.add(leftButton!)
28
29 splitPane!.add(leftPanel!, javax.swing.JSplitPane.LEFT)
30 splitPane!.add(rightButton!, javax.swing.JSplitPane.RIGHT)
31 splitPane!.validate()
32
33 process_events
```

Figure 7. Place the BBjControl on a JPanel (line 26)

continued...

Run the code `ClientObj-Fig7.src` shown in **Figure 7** and adjust the splitter of the JSplitPane. Notice the BBjButton changes size, as illustrated in **Figure 8**.

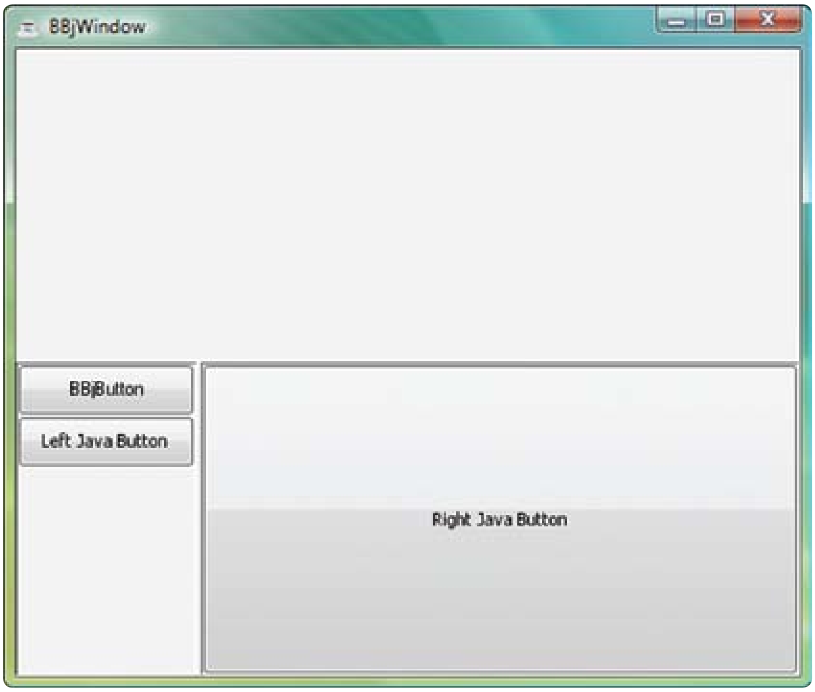
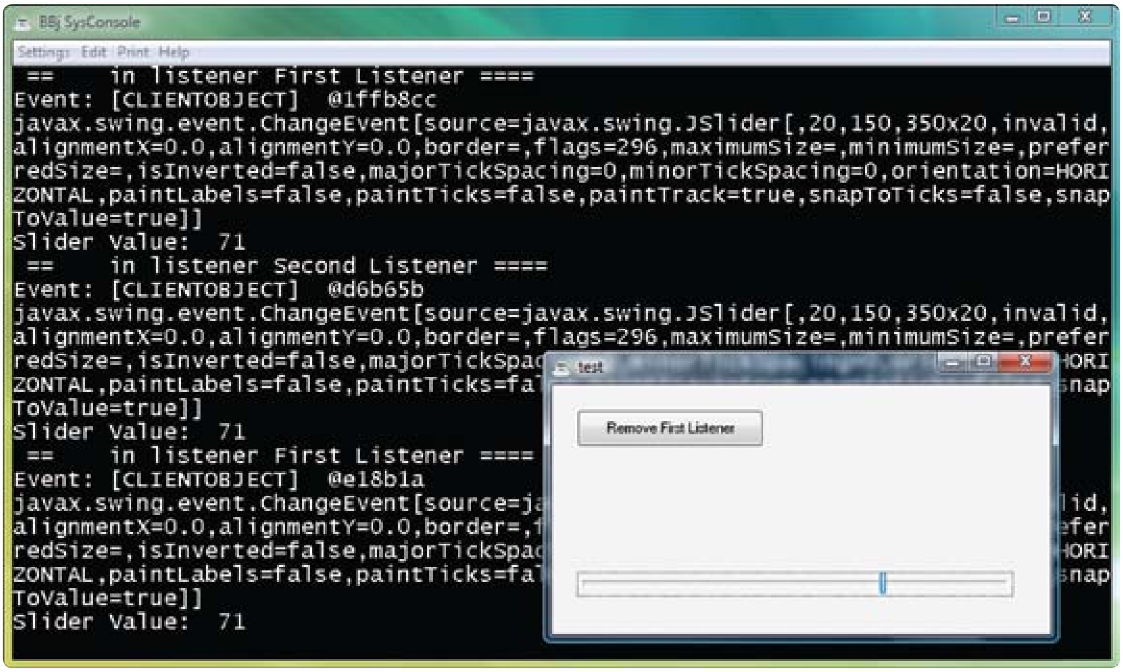


Figure 8. The result of running the code in Figure 7

Responding to Java Events Within a BBj Program

Most Java GUI components generate events in response to user actions. A BBj program can receive events that client-side Objects generate and the program can respond to those events using BBj code. The BBj code can manipulate the GUI controls or it may respond by only executing more traditional BBj code. In order to respond to events, a BBj program must use CustomObjects.

To see how to place a JSlider onto a BBjWindow and listen for the events generated by the JSlider, download the samples listed at the end of this article and run `ClientObj-Fig9.src`. Running this example prints the ChangeEvent and the value of the slider each time the user moves the slider.



```
 ==    in listener First Listener ====
Event: [CLIENTOBJECT]  @1ffb8cc
javax.swing.event.ChangeEvent[source=javax.swing.JSlider[,20,150,350x20,invalid,
alignmentX=0.0,alignmentY=0.0,border=,flags=296,maximumSize=,minimumSize=,prefer
redSize=,isInverted=false,majorTickSpacing=0,minorTickSpacing=0,orientation=HORI
ZONTAL,paintLabels=false,paintTicks=false,paintTrack=true,snapToTicks=false,snap
ToValue=true]]
Slider Value:  71
 ==    in listener Second Listener ====
Event: [CLIENTOBJECT]  @d6b65b
javax.swing.event.ChangeEvent[source=javax.swing.JSlider[,20,150,350x20,invalid,
alignmentX=0.0,alignmentY=0.0,border=,flags=296,maximumSize=,minimumSize=,prefer
redSize=,isInverted=false,majorTickSpac                                     HORI
ZONTAL,paintLabels=false,paintTicks=fa                                      snap
ToValue=true]]
Slider Value:  71
 ==    in listener First Listener ====
Event: [CLIENTOBJECT]  @e18b1a
javax.swing.event.ChangeEvent[source=ja                                     lid,
alignmentX=0.0,alignmentY=0.0,border=,f                                     efer
redSize=,isInverted=false,majorTickSpac                                     HORI
ZONTAL,paintLabels=false,paintTicks=fa                                      snap
ToValue=true]]
Slider Value:  71
```

Figure 9. Print the ChangeEvent and the value of the slider

continued...

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

Realizing the Benefit of ClientObjects

ClientObjects allow the BBj developer to take advantage of Java components that the open-source community and in-house Java developers have written. The integration of these third party controls into a BBj application is a seamless extension of the more traditional use of embedded Java. The RMI functionality of ClientObjects allows manipulation of the third party control in essentially the same way as within a native Java application. This integration allows the developer to mix-and-match the BBj controls, data structures, and other business-oriented features of BBj seamlessly with Java GUI controls as well as other non-GUI Java classes.

Registering ClientObject Jars for Deployment

BBj expects to find the Java class files for client-side objects in jar files that either are jar files in the JDK or are registered jar files. If a ClientObject references a class that is not part of the JDK, then BBj will check the jar file that contains the class to see if the jar file is a registered jar. If the jar is not registered, then BBj will display a nag message to the user asking to place the class into a registered jar.

There are two ways to register a jar; by using the executable jar BASIS provides or by executing a Java command.

To use the executable jar, enter the following command at the command prompt after setting the appropriate classpath:

```
java com.basis.jarreg.client.JarRegistrar inputJarName outputJarName
```

In order to register a jar from within BBj or from within a Java program, execute the following command

```
com.basis.jarreg.client.JarRegistrar.registerJar (inputJarName, outputJarName)
```

These commands create a registered copy of the input jar file. The newly created copy contains additional information in the `META-INF/BASIS.KEY` file that BBj uses in order to recognize that this new jar file is a registered jar. If anyone modifies the content of a registered jar, the registration becomes invalid and the jar must be re-registered.

When a jar is registered, the JarRegistrar first establishes an internet connection to BASIS to determine the current version of BBj that is available from BASIS. Next, it places that version number in the file META-INF/BASIS.KEY within the output jar. Registered jars are forward compatible; registered for all present and future BBj versions. This means, for example, that a developer registers a jar during 2009 when the current license version is 9.xx, then that jar is recognized as a registered jar so long as BBjServices runs with a 9.0 license or greater. If BBjServices uses an 8.0 license, then it will display a nag message.

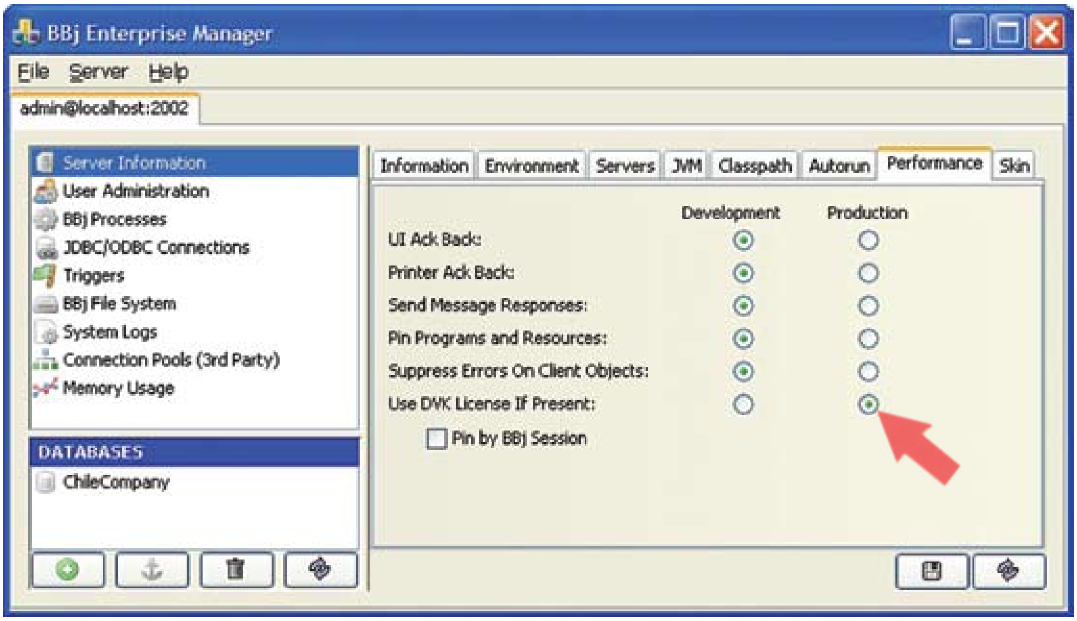
Developers do not need to use registered jars during development if they are running with a DVK (Developer's Version Kit) license. When running with a DVK license, available from BASIS Sales, BBj uses any Java class as a ClientObject without generating a nag message.

Testing and Deploying a Registered Jar File

When developing an application in an environment with a DVK license, it is very important to test the application with the DVK feature disabled prior to deploying the application. To run BBj with the DVK license disabled, follow theses steps.

1. Log in to BBj Services using the Enterprise Manager.
2. Select the **Server Information** item from the navigation list at the left side of the Enterprise Manager.
3. Click on the **Performance** tab in the right side informational area.
4. Choose the **Production** radio button for the **Use DVK License If Present** option shown in **Figure 10**.
5. Click on the [Save] button to save your new settings.

continued...



	Development	Production
UI Ack Back:	●	○
Printer Ack Back:	●	○
Send Message Responses:	●	○
Pin Programs and Resources:	●	○
Suppress Errors On Client Objects:	●	○
Use DVK License If Present:	○	●

□ Pin by BBj Session

Figure 10. Setting the DVK license for 'Production' mode

Once you are ready to deploy your application, make all your custom jar files available to the server as well as to the client. Add the jar files to the classpath of the server using Enterprise Manager by choosing the Classpath tab under the Server Information section. If you are using Web start to deploy to your clients, then simply add your custom jars in your file.

Summary

ClientObjects allow a BBj program to use arbitrary Java classes on the client. The RMI functionality of ClientObjects provides seamless integration of client-side objects into a BBj program. The BBj program can receive events that the client-side objects have generated and can invoke methods of the client-side objects in the same way that the program invokes methods of server-side objects.

ClientObjects are especially useful for including new GUI classes in a BBj application. Developers can mix-and-match third party GUI components with traditional BBjControls and place these third party controls on a BBjWindow. They can even add BBjControls to third party JComponents.

ClientObjects further reduce the boundary between BBj and Java, allowing developers to access the full power of both languages. They can use BBj for the things that BBj does best and Java for the things that Java does best. In addition, BBj developers are now in a position to take advantage of the large number of third party Java components that are available on the Web. They can integrate anything from a Web browser to a PropertySheet or a DigitalClock into a BBj program with a minimal development effort. ClientObjects, indeed, blow the doors of software development wide open.

Download the code samples from
www.basis.com/advantage/mag-v12n1/clientobjects.zip

Read more about Client Objects at
ClientObject Tutorial
www.basis.com/solutions/ClientObjectTutorial.pdf

BBj Custom Objects Tutorial
www.basis.com/solutions/BBj_CustomObjects.pdf

A Primer for Using BBj Custom Objects
www.basis.com/advantage/mag-v10n1/primer.html

† Download with BBj from www.basis.com/products/bbj/download.html and select "Demos" in the Optional File section. After completing the installation, select **BBj > Demos > LaunchDock** from the BASIS folder to run the demo.

The Extreme Enterprise Manager Makeover

An easy way to discover SQL optimizations!

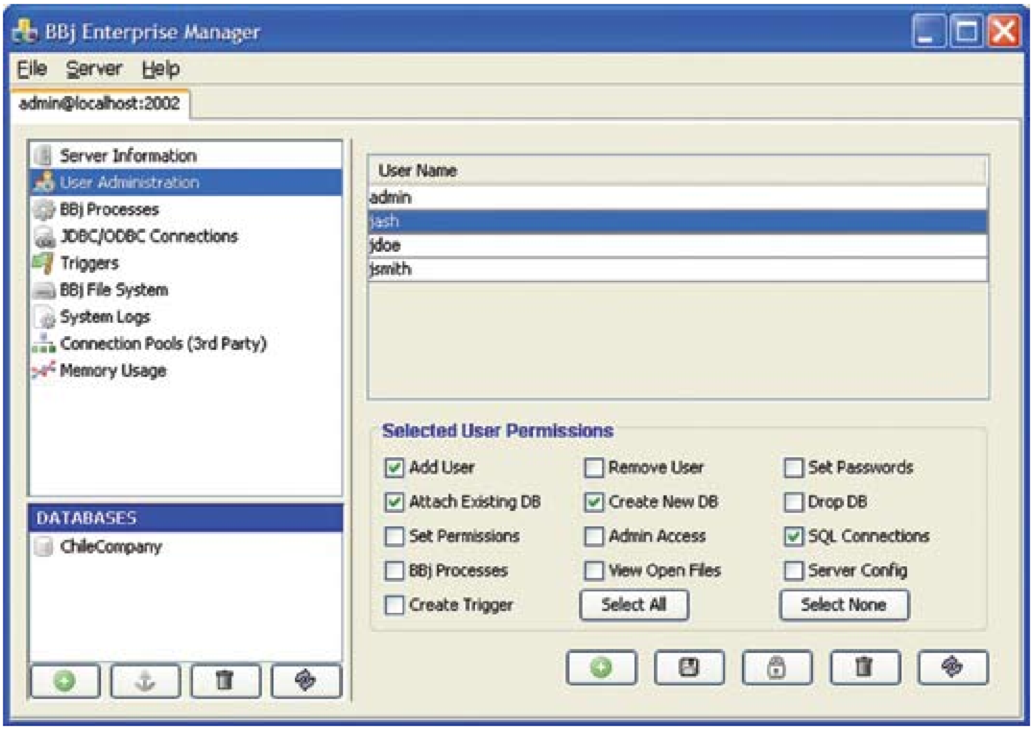
By Jeff Ash

Everyone who manages a BBj® installation will at one time or another use the BBj Enterprise Manager to administer users, set various configuration options, create/modify databases, view open files, terminate hung processes, and a host of other options. Since this component of the BBj system is so important to those managing their system, BASIS redesigned the user interface, made a number of improvements, and added a number of features to make the job of the system and database administrator more efficient.

The New Interface

The new user interface is intuitive, easy to use, and quick to navigate. It is no longer necessary to navigate through multiple levels of tree nodes to find the sought after information because everything is much more visible to the user. BASIS replaced the Data Dictionary Module in the BASIS IDE with the new Enterprise Manager interface, so that developers can manage both their databases and all server configuration settings right from the IDE. One of the greatest benefits is that there is now only one interface to learn since the IDE and the Enterprise Manager both share this new interface.

The new interface also boasts a number of "ease-of-use" features to simplify the system administrator's tasks. For example, the user management panel is cleaner, makes setting permissions for users easier, and now includes [Select All] and [Select None] buttons as seen in **Figure 1**.



User Name
admin
jash
jdoe
jsmith

Selected User Permissions

☑ Add User □ Remove User □ Set Passwords
☑ Attach Existing DB ☑ Create New DB □ Drop DB
□ Set Permissions □ Admin Access ☑ SQL Connections
□ BBj Processes □ View Open Files □ Server Config
□ Create Trigger

Figure 1. New Enterprise Manager interface

Jeff Ash
Software Engineer

In addition, the trigger-editing dialog uses color coding and differing fonts to show the types of triggers currently configured on a file as well as the enabled/disabled state of each trigger. There are also a number of helpful dialogs with explanations or warnings when invoking specific user actions. The Enterprise Manager now offers tooltips on nearly every button.

continued...

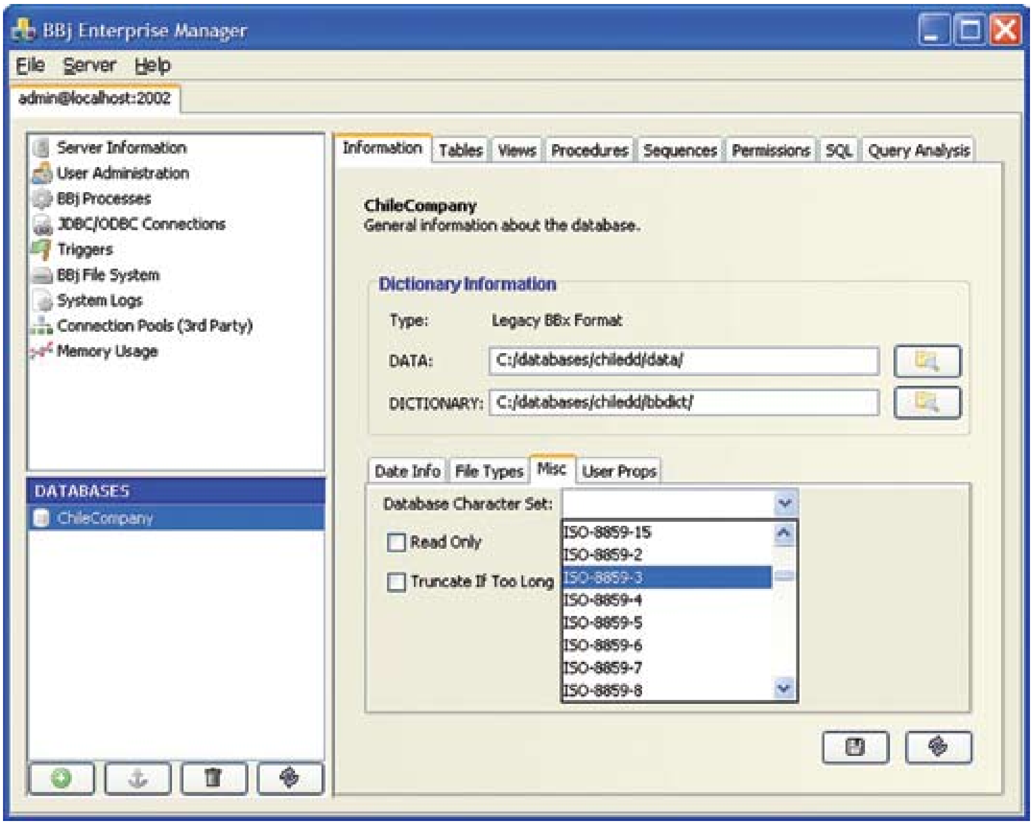
Database Administration

One of the most common uses of the Enterprise Manager is for managing and configuring databases. The new Enterprise Manager includes a number of new features and improvements to existing features, which make this job more efficient.

Configuration Features

An administrator can now specify up to three different date formats for interpreting date values in their database tables. While it is highly discouraged to use more than one date format for storing dates in a database, it is not uncommon to find older applications that use this technique.

Since a worldwide development community creates applications with BBj, administrators can now select the character set most appropriate for their local language to use when reading and writing data to a database. The database configuration panel, shown in **Figure 2**, displays a list of the available character sets on the server.



ChileCompany
General information about the database.

Dictionary Information

Type: Legacy BBx Format

DATA: C:/databases/chiledd/data/

DICTIONARY: C:/databases/chiledd/bbdict/

Database Character Set:

ISO-8859-15
ISO-8859-2
ISO-8859-3
ISO-8859-4
ISO-8859-5
ISO-8859-6
ISO-8859-7
ISO-8859-8

☐ Read Only

☐ Truncate If Too Long

Figure 2. Database Configuration Panel

Database Management Features

In BBj 8.0, the Enterprise Manager introduces a host of new and/or improved features for managing the actual data and structure of BBj databases. The following list highlights some of the most prominent features:

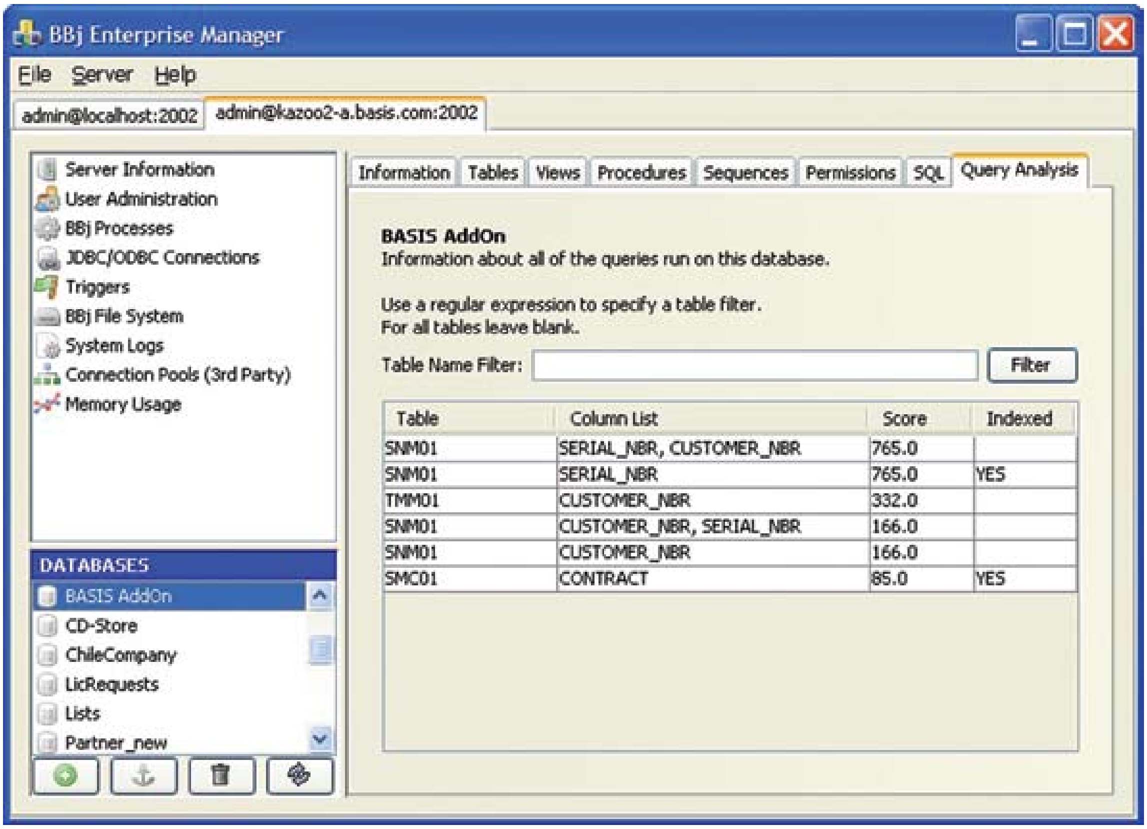
- Create and manage ESQL tables (the new SQL-only file format in BBj that supports transactions, integrity constraints, and more)
- Create and drop Sequences and view the current value of the Sequence
- Change the file type of a table by selecting the type from a simple drop-down list, and automatically convert and update the file
- Generate a string template from the column definitions of a table
- Type or copy/paste a string template and generate the column definitions for a table automatically based on the template contents
- Add/remove/modify columns on a table and update the record layout and records of the physical data file automatically to match the new definition

continued...

Query Analysis

It is often difficult to track down specific performance issues in SQL queries since these statements can be arbitrarily complex, creating the potential for numerous combinations of tables and columns. Determining which columns to index and the order the segments of the index should follow can be a difficult task. BASIS added a new feature - Query Analysis - to the SQL engine and the Enterprise Manager that removes the guesswork from this task.

The SQL engine automatically gathers information during execution of every SQL query regarding the columns included in the WHERE clause of each query. The Enterprise Manager retrieves this information and allows the database administrator (DBA) to analyze results such as those shown in **Figure 3**.



BASIS AddOn
Information about all of the queries run on this database.

Use a regular expression to specify a table filter.
For all tables leave blank.

Table Name Filter:

Table	Column List	Score	Indexed
SNM01	SERIAL_NBR, CUSTOMER_NBR	765.0	
SNM01	SERIAL_NBR	765.0	YES
TMM01	CUSTOMER_NBR	332.0	
SNM01	CUSTOMER_NBR, SERIAL_NBR	166.0	
SNM01	CUSTOMER_NBR	166.0	
SMC01	CONTRACT	85.0	YES

Figure 3. Sample Query Analysis

To access the Query Analysis information, select the database from the Databases list, then click on the Query Analysis tab. Each row shows the name of the table and a combination of columns in a specific order. The columns refer to the columns that were included in the WHERE clause of at least one SQL statement. There is a row for each possible permutation of the order of the columns. The score refers to the frequency of the combination of columns used in SQL statements and is a relative number. The final column shows “YES” if this specific combination of columns is currently indexed.

DBAs can use this information to help determine what, if any, indexes to create on a database to improve performance of their queries. As illustrated in **Figure 3**, users often employ a combination of SERIAL_NBR and CUSTOMER_NBR on the SNM01 table in their queries since it is at the top of the list. However, there is no index on this combination. This could potentially be a candidate for a new index.

The job of the DBA is to determine when to create an index to avoid creating indexes on every combination possible. For example, the database in the example above has an index on SERIAL_NBR. Knowledge of the database structure shows that there really is no need for another index on SERIAL_NBR and CUSTOMER_NBR because there should only be one CUSTOMER_NBR for each SERIAL_NBR.

Summary

BASIS continues to improve the productivity and experience of its end users and developers. The new Enterprise Manager interface and its availability in the BASIS IDE takes another giant step in this direction. DBA’s will certainly enjoy the new database configuration options and table design capabilities. They can use the new Query Analysis information to find potential performance bottlenecks in SQL statements to make queries fly. With this new and cleaner interface, administrators can manage their tasks faster and with greater ease.

AddonSoftware

AddonSoftware™ is a cost-effective, modular, full-featured, and fully integrated enterprise resource planning solution for mid-market industries.

• Accounting • Distribution • Manufacturing

Revitalized and completely renovated, AddonSoftware demonstrates the synergistic power of BBj® and Barista® in providing vertical market application developers with the building blocks for their own Barista applications!

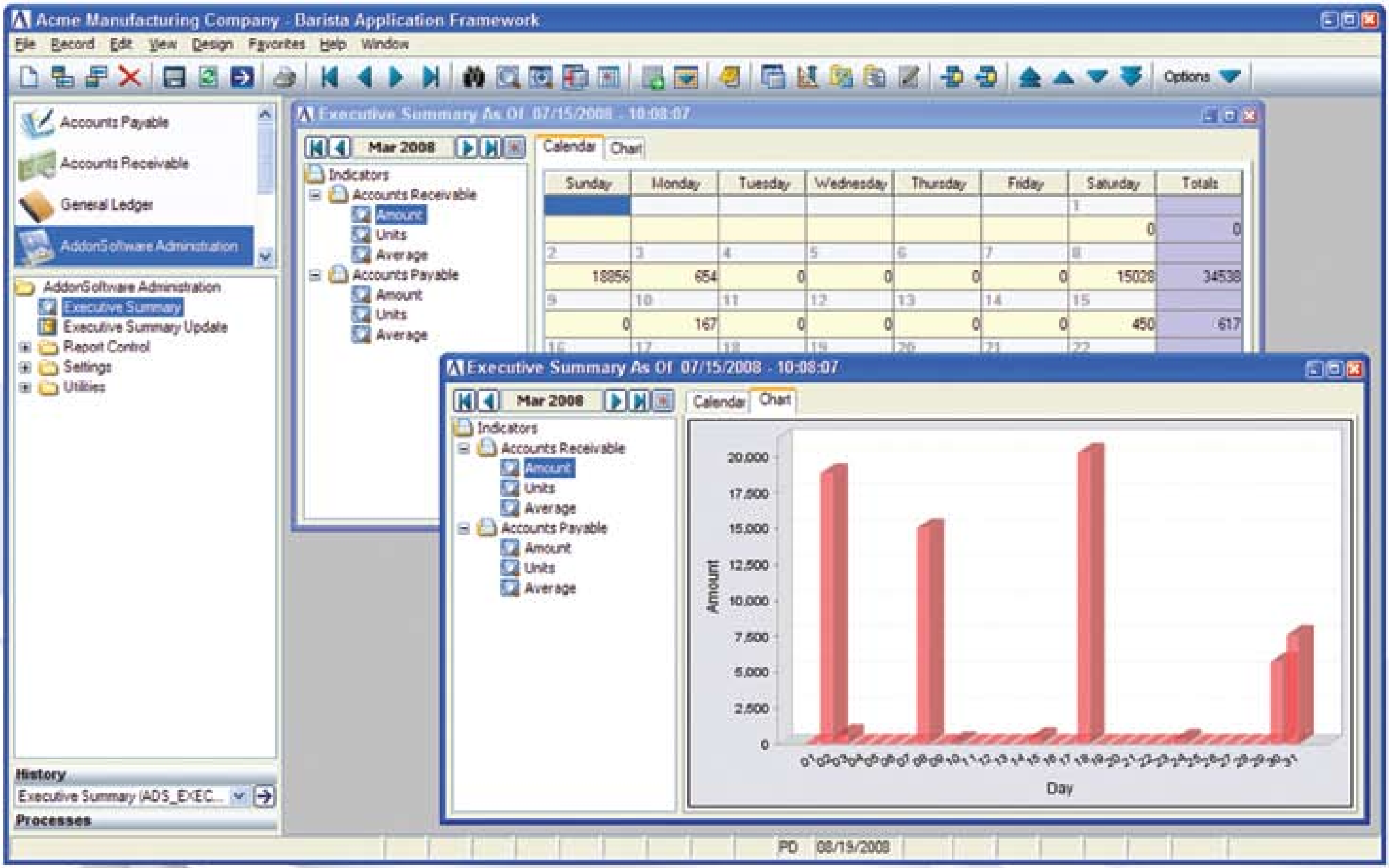
Modern - Keeps pace with current GUI look and feel standards

Productive - Efficient custom development using the Barista tool

Extensible - Incorporates commercial or open source Java libraries

Flexible - Delivers cross-platform GUI - Windows, Linux, Mac, UNIX

Trusted - Builds on decades of reliability


Acme Manufacturing Company - Barista Application Framework
File Record Edit View Design Favorites Help Window
Options
Accounts Payable
Accounts Receivable
General Ledger
AddonSoftware Administration
History
Processes
Amount
Day
20,000
17,500
15,000
12,500
10,000
7,500
5,000
2,500
0

AddonSoftware Administration
- Executive Summary
- Executive Summary Update
- Report Control
- Settings
- Utilities

Executive Summary (ADS_EXEC...

Executive Summary As Of 07/15/2008 - 10:08:07

Mar 2008

Calendar | Chart

Indicators
- Accounts Receivable
  - Amount
  - Units
  - Average
- Accounts Payable
  - Amount
  - Units
  - Average

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Totals
						1	
						0	0
2	3	4	5	6	7	8	
18856	654	0	0	0	0	15028	34538
9	10	11	12	13	14	15	
0	167	0	0	0	0	450	617
16	17	18	19	20	21	22	

PD 08/19/2008

Unleashing *The Freedom to Grow*®

U.S. 1.800.423.1394 • International +1.505.338.4188

www.addonsoftware.com

AddonSoftware™

Quick and Easy Solutions With Free Java Libraries – *Part I*

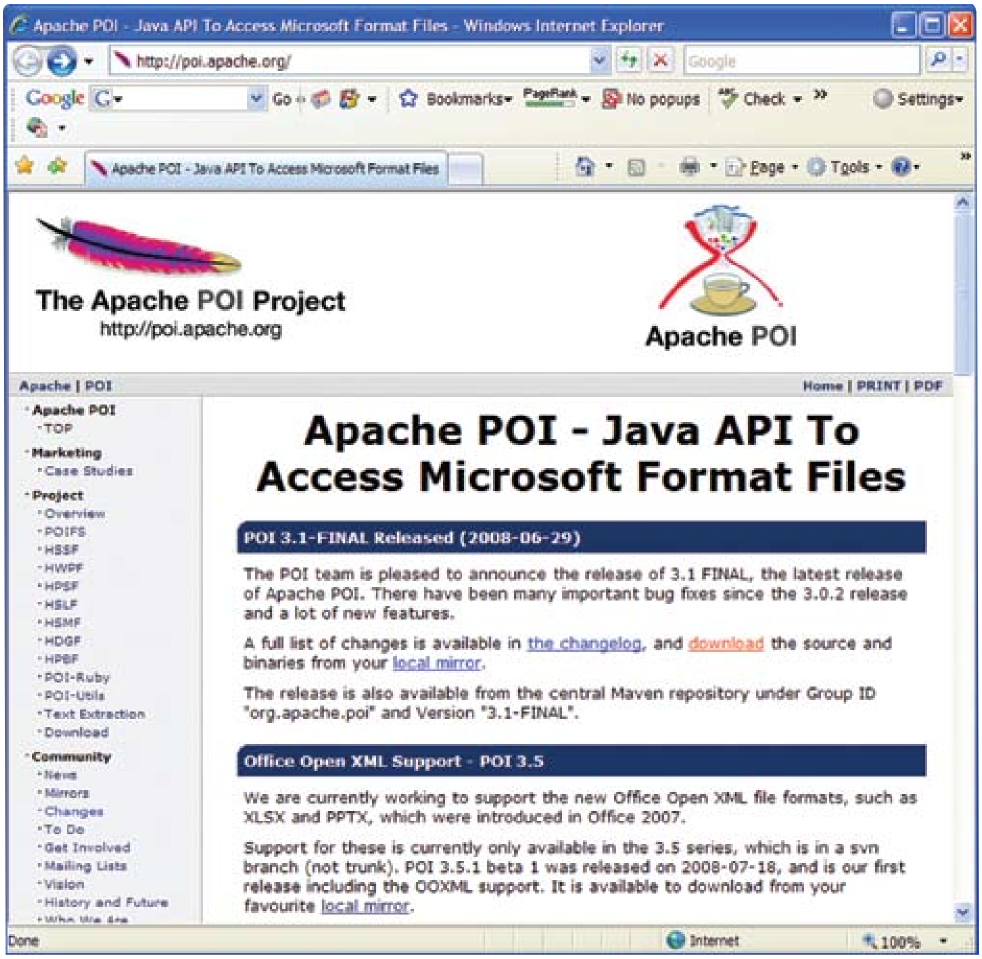
By Shaun Haney

The BBj® API is a toolkit full of solutions for everyday business needs. Businesses of varying sizes may have the need to integrate charts into applications, map data from a database to human-readable information, or even transfer files between client and server. These very viable requirements are now extremely easy to deliver.

While BASIS continues to add to the breadth of the API, it is possible that the ready-made solutions developers are looking for do not yet exist within the BASIS API. This does not mean they must develop the entire solution themselves. BBj allows developers to integrate Java objects into their application, and with BBj Client Objects, they can perform the work on the client's side.

Say, for example, you are a business with several sales offices that needs an intranet application to convert a customer query from your BBj databases instantly to an Excel spreadsheet. There are, of course, several ways of achieving this task. One solution is to use the BBj ODBC Driver with Microsoft Excel, however it requires that the users have Microsoft Excel and ODBC configured on their machine.

Perhaps a better solution would be to use a Java library that reads and writes Excel files (see **Figure 1**), and then creates the Excel files programmatically. Your system administrator could use JNLP (Java Network Launch Protocol) to deploy the BBj program that performs this task, thus making the program easily accessible via a URL on the business' internal Web site. Users could create the Excel files on their own machines (instead of the server) using BBj's new ClientObjects. While the client systems must have Java installed, they do not require Microsoft Excel or ODBC to create the Excel document on their machine.


Apache POI - Java API To Access Microsoft Format Files - Windows Internet Explorer
http://poi.apache.org/
Google
Google
Go
Bookmarks
PageRank
No popups
Check
Settings
Apache POI - Java API To Access Microsoft Format Files
Page
Tools
Apache POI
Done
Internet
100%

The Apache POI Project
http://poi.apache.org

Apache | POI — Home | PRINT | PDF

- Apache POI
  - TOP
- Marketing
  - Case Studies
- Project
  - Overview
  - POIFS
  - HSSF
  - HWPF
  - HPSF
  - HSLF
  - HSMF
  - HDGF
  - HPBF
  - POI-Ruby
  - POI-Utils
  - Text Extraction
  - Download
- Community
  - News
  - Mirrors
  - Changes
  - To Do
  - Get Involved
  - Mailing Lists
  - Vision
  - History and Future
  - Who We Are

Apache POI - Java API To Access Microsoft Format Files

POI 3.1-FINAL Released (2008-06-29)

The POI team is pleased to announce the release of 3.1 FINAL, the latest release of Apache POI. There have been many important bug fixes since the 3.0.2 release and a lot of new features.

A full list of changes is available in the changelog, and download the source and binaries from your local mirror.

The release is also available from the central Maven repository under Group ID "org.apache.poi" and Version "3.1-FINAL".

Office Open XML Support - POI 3.5

We are currently working to support the new Office Open XML file formats, such as XLSX and PPTX, which were introduced in Office 2007.

Support for these is currently only available in the 3.5 series, which is in a svn branch (not trunk). POI 3.5.1 beta 1 was released on 2008-07-18, and is our first release including the OOXML support. It is available to download from your favourite local mirror.

Figure 1. Example of a Java library

Searching the Internet for this or other such Java libraries is relatively easy. Downloading and executing (integrating) such a library may require a bit more guidance so this article continues with an in depth step-by-step guide to downloading and utilizing a library that reads and writes Excel files. With BBj's new ClientObjects, the world is your oyster - you can do anything you want!

Shaun Haney
Quality Assurance Engineer

There is more!
Follow along with Part 2 at
www.basis.com/advantage/mag-v12n1/libraries2.pdf

BASIS Web Training is Lean and Green

By Amer Child

For over a year, BASIS has conducted a variety of Web-based training courses. In 2008 alone, BASIS offered 18 courses to developers interested in growing their businesses. Developers from the US as well as Canada, Mexico, and Puerto Rico to as far away as Australia joined us for Web training, saving them thousands of dollars and days out of the office. Sitting comfortably at their own desks in their own cities, they received technical updates and hands-on training from the experts at BASIS. The results and feedback have been positive.

> *"The Web based training was so great! I am sold on this style of training. Even with the occasional pause for helping those that had fallen behind, it worked out as I was able to stay up with other tasks that I would not have been able to do in an actual classroom setting. Furthermore, this allowed me to be in the environment in which I actually work so it felt more natural to continue using what I learned."*
>
> — Mark Robinson, CFO/CTO, Auto-Rain Supply, Inc.

Using Web-based training, BASIS is able to bring experts from any part of the globe to attendees anywhere in the world. Only our most knowledgeable and experienced professionals conduct the training.

Have you checked our training calendar recently? Go to www.basis.com/events/training

We are sure there are topics that can improve your organization's revenue in these tough economic times. Save fuel, save time, save money and go lean and green. See you on the Web!

Barista Habla Español

By Amer Child

In late September, BASIS and ESS, the distributor for BASIS products to most of Latin America, visited customers in San José, Costa Rica. These customers were treated to in-depth product information and company insights during the three-day event, highlighted by Barista®, BASIS' new rapid application development framework.

On the first day, Nico Spence (BASIS) teamed up with Jésus Alvarez and William Hernandez (ESS) to present the BASIS company and product roadmap; a technology update and view of "Barista in Action." Nico made excellent use of the simultaneous language translation services in the session to engage in an English/Spanish dialog with the twenty plus attendees.

During the two days that followed, William taught a Barista training, using the Spanish version of Barista, thereby quenching the thirst in the Costa Rican BASIS community for a working knowledge of this great new multilingual tool.

Based on the mounting interest and visible excitement among the attendees, ESS plans to host more Barista events in Spanish. Look out for an invitation to a Webinar in which Barista habla Español!

Photographs courtesy of Erick Garro

Presenting Barista in Costa Rica

Amer Child
Training & Sales Support Specialist

BASIS Adds Mapping Functionality to BBjRecordSets

"Suddenly, there came a RecordSet Mapping, Mapping at my Chamber Door!"

By Jon Bradley

Databound BBj controls make it easy to display and modify data in a GUI application. Using AppBuilder's FormGen Wizard, it is possible to create an entire file maintenance application without writing a single line of code. New in BBj 8.0, mapped recordsets make it possible to customize and enhance such an application easily.

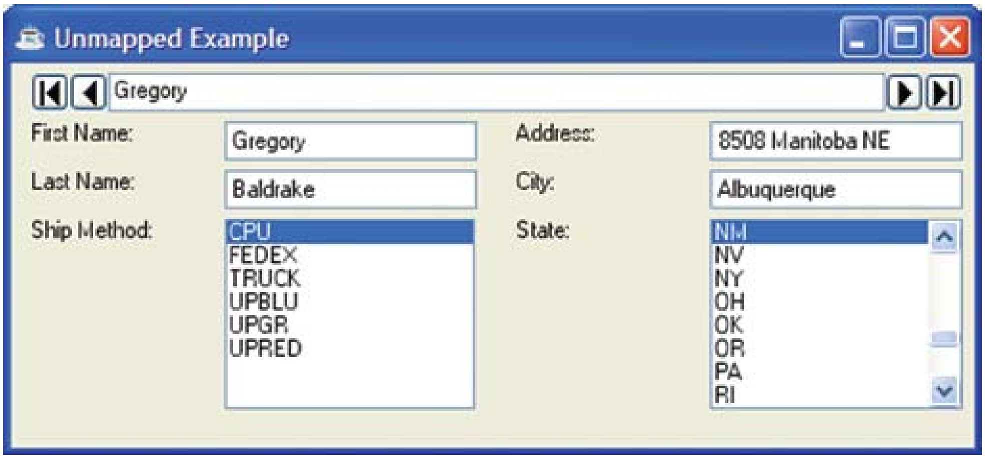
Trouble can rear its ugly head when displaying stored data. Typically, this data is condensed for storage purposes and therefore not in a format or expression familiar to the user. BBj 8.0 addresses this issue by introducing the ability to add display translations or "mappings" to your BBjRecordSets.

For example, Julian dates are stored in data files in a numerical format such as `2454628`. Most users would not be able to easily translate that date so the BBjInputD control provides the mapping to display `2454628` as `June 10, 2008`. Likewise, you could display two-character state codes as their full name where NM becomes New Mexico.

Mapping in Action

Figure 1 and **Figure 2** show the same basic application, however, **Figure 2** uses the new mapping functionality to change the data in the CUSTOMER table of our Chile Company database into a more user-friendly form. We used the STATE table to display the corresponding full name for the two-character state codes stored in the CUSTOMER table. We perform a similar trick to convert the preferred shipping method from the five-character code stored in the CUSTOMER table to the full description.

continued...



Unmapped Example

Gregory

First Name: Gregory

Last Name: Baldrake

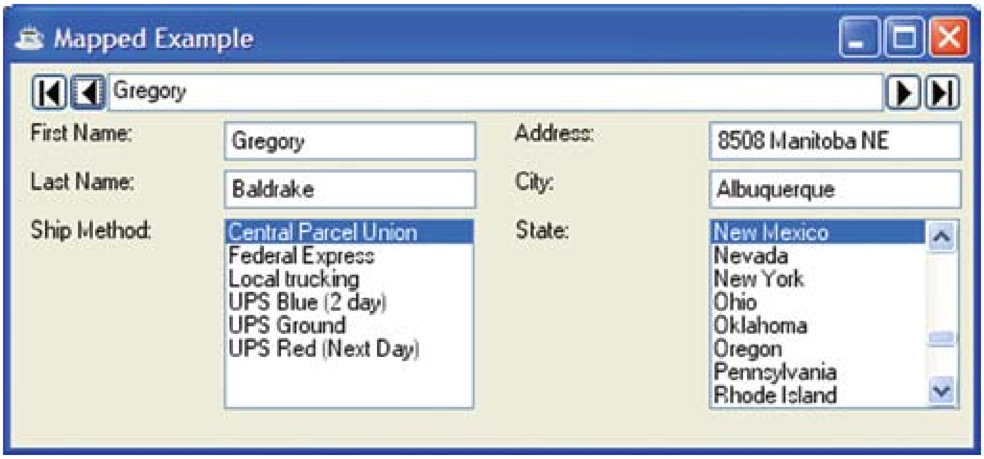
Ship Method: CPU, FEDEX, TRUCK, UPBLU, UPGR, UPRED

Address: 8508 Manitoba NE

City: Albuquerque

State: NM, NV, NY, OH, OK, OR, PA, RI

Figure 1. Application shows the raw data in the database



Mapped Example

Gregory

First Name: Gregory

Last Name: Baldrake

Ship Method: Central Parcel Union, Federal Express, Local trucking, UPS Blue (2 day), UPS Ground, UPS Red (Next Day)

Address: 8508 Manitoba NE

City: Albuquerque

State: New Mexico, Nevada, New York, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island

Figure 2. Application shows the meaningful data by using recordset mappings

Jon Bradley
Software Engineer

Furthermore, when the user changes the state or shipping method in the GUI front end, the data converts automatically from the full name back to the compact two-character state code. Before the addition of the mapping functionality, the application in **Figure 2** would have required either storing the full descriptions in the CUSTOMER table or writing the many lines of code required to handle the retrieval, mapping, display, and modification of the data.

Shut the Door on Database Changes

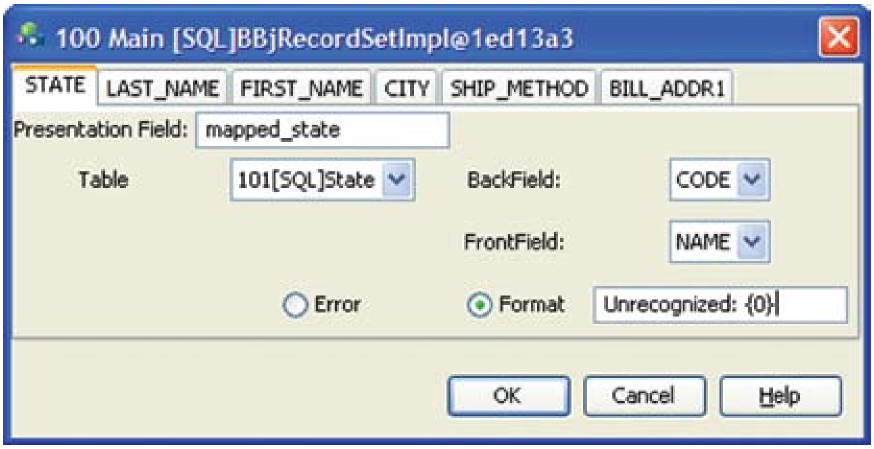
Changing the database, the alternative to using recordset mappings, is a bad solution. Since databases use foreign keys to reduce size, increase data normalization, and increase referential integrity, throwing them out just to display the description corresponding to that key is not a worthwhile consideration. Mappings provide an easy way to display the human readable descriptions without adding cumbersome code or denormalizing your database.

Open the Door to Mapping

Adding mappings to your application is simple; it introduces a new field to the recordset that becomes the field name for data binding a control.

A mapping consists of the value source and the mapping action. The value source is typically a BBjRecordSet that specifies how the database values should be mapped to presentation values and vice versa. The mapping action specifies what to do when the value source does not provide a valid mapping for a value.

The Chile Company database already contains a table STATE, which maps two-character state codes to full names. The database also has a table SHIP_METHOD that maps the five-character shipping codes to their full names. FormBuilder provides a dialog (**Figure 3**) to add mappings to a BBjRecordSet, the data from which the FormGen Wizard can then create the application. BASIS also expanded the API with new methods to add mappings.



100 Main [SQL]BBjRecordSetImpl@1ed13a3

STATE | LAST_NAME | FIRST_NAME | CITY | SHIP_METHOD | BILL_ADDR1

Presentation Field: mapped_state

Table: 101[SQL]State

BackField: CODE

FrontField: NAME

○ Error ● Format Unrecognized: {0}

Figure 3. Mapping Editor Dialog in FormBuilder

If the number of possible choices for a field is relatively small, you may choose to build an in-memory BBjRecordSet to serve as your mappings.

Given a file like this:

Employee Table

NAME	ADJECTIVE CODE	HAS CAR CODE	JOB CODE
Jon	A	Y	1
Dave	B	Y	2
Tom	C	N	3
Frank	A	N	4

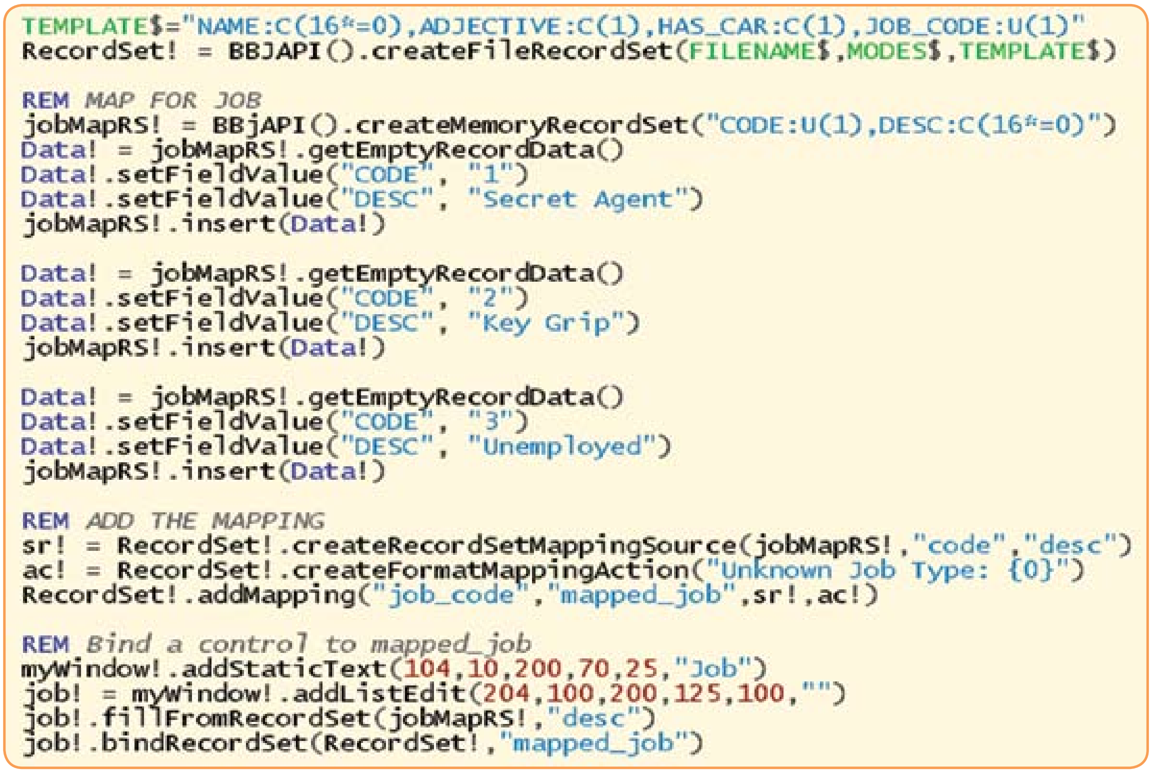
Where the Job Codes stand for the following:

Job Code Table

JOB CODE	JOB DESCRIPTION
1	Secret Agent
2	Key Grip
3	Unemployed
4	(NO MAPPING)

continued...

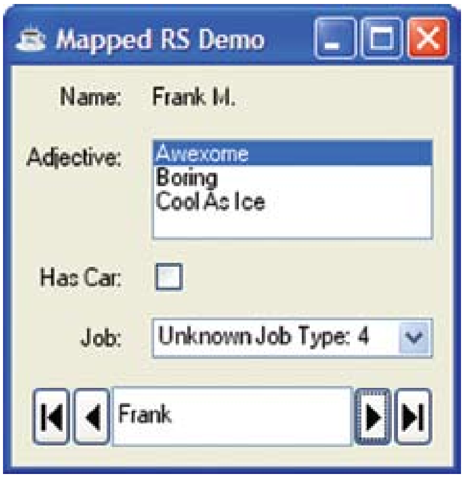
The code sample excerpt in **Figure 4** (download the complete sample file `MappedRecordset.src` noted at the end of this article) creates a mapping and binds a control to the new mapped adjective field.



```
TEMPLATE$="NAME:C(16*=0),ADJECTIVE:C(1),HAS_CAR:C(1),JOB_CODE:U(1)"
RecordSet! = BBJAPI().createFileRecordSet(FILENAME$,MODES$,TEMPLATE$)

REM MAP FOR JOB
jobMapRS! = BBjAPI().createMemoryRecordSet("CODE:U(1),DESC:C(16*=0)")
Data! = jobMapRS!.getEmptyRecordData()
Data!.setFieldValue("CODE", "1")
Data!.setFieldValue("DESC", "Secret Agent")
jobMapRS!.insert(Data!)

Data! = jobMapRS!.getEmptyRecordData()
Data!.setFieldValue("CODE", "2")
Data!.setFieldValue("DESC", "Key Grip")
jobMapRS!.insert(Data!)

Data! = jobMapRS!.getEmptyRecordData()
Data!.setFieldValue("CODE", "3")
Data!.setFieldValue("DESC", "Unemployed")
jobMapRS!.insert(Data!)

REM ADD THE MAPPING
sr! = RecordSet!.createRecordSetMappingSource(jobMapRS!,"code","desc")
ac! = RecordSet!.createFormatMappingAction("Unknown Job Type: {0}")
RecordSet!.addMapping("job_code","mapped_job",sr!,ac!)

REM Bind a control to mapped_job
myWindow!.addStaticText(104,10,200,70,25,"Job")
job! = myWindow!.addListEdit(204,100,200,125,100,"")
job!.fillFromRecordSet(jobMapRS!,"desc")
job!.bindRecordSet(RecordSet!,"mapped_job")
```

Figure 4. Excerpt from `MappedRecordset.src`

In **Figure 5**, the job field displays a formatted message, as specified by the mapping action, because the value source does not specify a mapping for the value 4.



Name: Frank M.

Adjective: Awexome
Boring
Cool As Ice

Has Car: □

Job: Unknown Job Type: 4

Frank

Figure 5. The programmatic example application

Summary

The new mapping functionality makes it easier to use BBjRecordSets in normalized databases. The mappings make displaying human readable descriptions of table data easy, whether it is done in AppBuilder or via the BBj API. BASIS continues to make improvements in the mapping system and encourages developers to use the latest release to get the most functionality and speed from this new feature. So give mapping a warm welcome into your applications.

Download the code sample at
www.basis.com/advantage/mag-v12n1/mappedrecordset.zip

For more information about recordset mappings, read
Working With RecordSet Mappings
www.basis.com/solutions/RecordSetMappings.pdf

BBj Thrives on the Web

By Stephan Wald

Worldwide, far more than 1,000,000 items are ordered every year over e-commerce sites driven by BASIS products. Good portions of these orders are processed through Web shops provided by BASIS Europe. When BASIS incorporated the acquisition of PHAROS Ltd., a former BASIS partner in Europe, to the BASIS Europe portfolio in 2006, BASIS inherited PharoShop, a solution written in BBj® and PRO/5®. This solution, delivered through the Internet, integrates with third party software packages as well as software packages written in Business BASIC. Renamed BASIS Europe Web Shop, it is deployed in five different European countries and runs in four different languages – a truly pan-European solution.

The Concept is Versatile

BASIS Europe offers a solution with complete functionality to meet its customers' broad spectrum of needs. Basic functionality might start out as a powerful full text retrieval function, offering categorized browsing of items and shopping cart order processing, then evolve into a sophisticated SAP OCI interface, a shop-in-shop integration (facility for subordinated resellers such as a POS-kiosk), an interface to eBay, a compatible database for printer supplies, a reader for mobile USB barcode scanners, or even a complex result set ranking. Regardless of the level of complexity, all of these functions are implemented using the powerful Business BASIC language from BASIS.

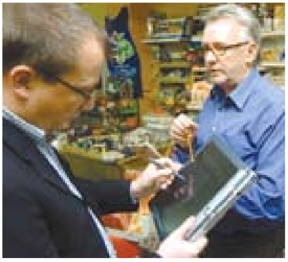
The Technical Foundation is BBj

The toolset for the product is BBj itself, utilizing an object-oriented Web API for the creation of HTML pages. This component was recently rewritten in object-oriented Business BASIC to meet clients' current expectations of modern software development. Though the first version – dating back to 1999 – was based on the CGI-based concept of the BASIC Web Utility, the current product has nothing in common anymore with those roots. The layout of the HTML is entirely separate from the code, allowing for highly flexible and efficient development. The object-oriented API brings the HTML template to life and generates the dynamic output for the user's Web browser.

Developers can use the BBj Web API from BASIS Europe to build any kind of Web application that ranges from online shops through customer care portals to entire Web-based applications using modern Ajax functionality to enhance the user experience.

BBj is #1 in Wholesale Stationery and Office Supplies in Germany

In addition to installations for wholesaling chemical goods, automotive parts, and spare parts for electrical installations, the BBj-based solution is the leading product used in the office supply and stationery industry. Of all the German wholesalers in office supplies and stationery, 80% use the shop system for their e-commerce. This is a direct result of the widespread use of Business BASIC in Germany since the late 1980's.



The Speed is the Key to the Success

"The performance and the flexibility of this system is amazing," says G. Vricke, IT manager at a BASIS customer in Belgium. He continues, "Rarely are Internet shops handling such an amount of data this fast." In fact, average installations of BASIS Europe Web Shops represent a product portfolio of at least 25,000 items – the largest one contains 825,000 different items – all made available for full text retrieval within a fraction of a second using Business BASIC keyed files for indexing. Written in BBj, the shop system not only uses data imported from interfaces to or stored in SQL databases, but it also takes advantage of lightning fast direct access to Business BASIC data files, hence facilitating the online access to any kind of data stored in any legacy Business BASIC application.

New Building Blocks for Business BASIC Programmers

The BASIS toolset is not specifically designed for the Web. Nonetheless BASIS Europe's Web Shop, written in BBj, demonstrates once again the power and flexibility of the programming language, delivering a powerful database-driven solution. There is no need to look beyond BBj in order to add Web-based functionality to existing Business BASIC solutions – you can stay in your familiar environment, reuse your experience, your data structures and even parts of your existing code.

For a demo, go to www.basis-europe.eu/demoshop

Stephan Wald
Director Sales and Technical Services

Partnership
Language/Interpreter
DBMS
Development Tools
System Administration
Applications

Freshly Brewed – *Barista Caffeinates Addon Accounting Applications!*

...Continued from page 9

Codeport performs functions such as removing line numbers, formatting code with indents, removing BEGIN statements and IOLISTS, replacing calls to legacy public programs with their more intuitive Version 8 names, and replacing common functions and routines with Version 8 counterparts. The developer makes further improvements by replacing GOTO's with structured code, using symbolic labels (e.g., `err=*next`) wherever possible, replacing references to obsolete sort files with alternate keys on the source files, and replacing IOLIST variable names with version 8 string template names, etc.

Dataport
The legacy AddonSoftware data was deficient in two major areas. Firstly, it carried the majority of the dates in a packed format. Secondly, it contained several non-normalized tables (one table containing multiple record types/formats). In addition, the legacy data dictionary contained several field names that were duplicated or were not intuitive. The Dataport utility is another AddonSoftware-specific tool intended to address these issues. Dataport unpacks all dates into YYYYMMDD format, separates old combined city/state address lines, replaces non-normalized files with individual files containing a single consistent record type, and performs several other conversions on AddonSoftware data. Dataport also makes use of several plain text files to specify file and/or field name changes between the legacy version and Version 8.

Import from BASIS Data Dictionary
Although Codeport and Dataport are tools developed specifically for upgrading AddonSoftware, Barista also includes a non-AddonSoftware specific tool for creating a Barista database. Developers who use the BASIS Data Dictionary (DD) in their applications are one step ahead in converting their applications to Barista. The Import from BASIS DD program will bring the BASIS DD across to Barista and, thereafter, any changes to the Barista dictionary will automatically be applied to the BASIS DD.

The import utility uses a configurable settings file. It reads the BASIS DD and incorporates these settings, creating Barista elements and table definitions. Again, with table definitions in place, a functional set of forms can result. In no time, the foundation of the application takes shape, running in Barista. Read *From the BASIS DD to a Barista App in a Flash* on page 12 of this issue.

Additional Barista Features

Document Management
After running the legacy report programs through the Codeport process and completing the post-Codeport cleanup, AddonSoftware developers had a functional set of reports. While these reports were adequate, they were limited just to the selected output device. If output was directed to the screen, the report appeared in the standard print preview window outside of the MDI. The developers decided to make another pass through those reports and convert them to use Barista's Document Output and Management System.

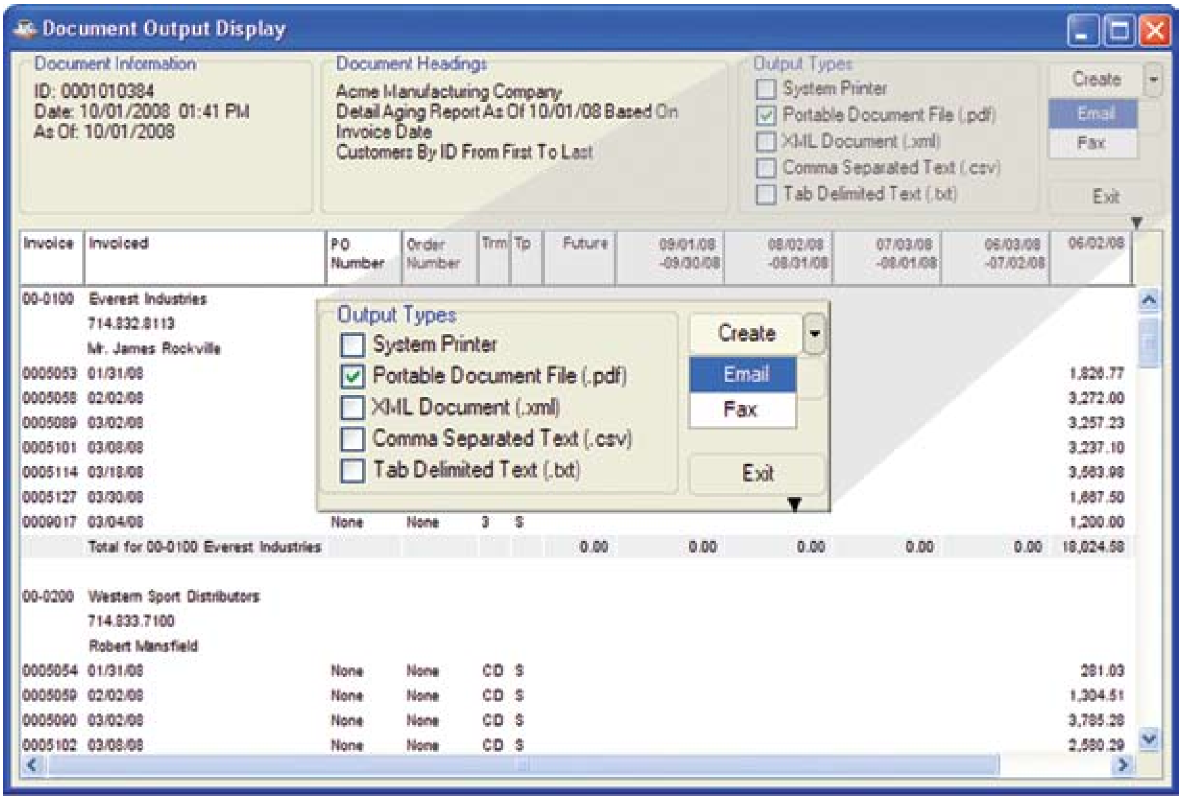
The Document Output and Management System, a.k.a. "DocOut," is another powerful component of Barista that offers several advantages over traditional printing. DocOut renders reports in a window within the MDI. The body of the report displays in a grid, the columns of which are user-adjustable. All reports rendered in DocOut, therefore, have a standardized look and feel. Once the document is created, users can opt to save the report in one or more output formats, as **Figure 6** illustrates.

Document settings are customizable as well, so users can alter the font size and orientation, change the archive location, or create a report in one or more of the available formats without ever displaying the DocOut window. This last option is great for reports like registers or daily statistics because it does not require any intervention from the user, which eliminates the age-old problem of running an update, only to realize the printer jammed, or the report was lost!

Fax and E-mail
Barista also offers integrated fax and e-mail capability, made possible by BBj's ability to incorporate third party Java fax libraries. From inside the DocOut window and after selecting an output type such as PDF, simply generate the file or open a supplemental window to enter fax or e-mail information.

It was important to include the ability in AddonSoftware 8 to fax and/or e-mail such documents as invoices, statements, or

continued...



Document Information
ID: 0001010384
Date: 10/01/2008 01:41 PM
As Of: 10/01/2008

Document Headings
Acme Manufacturing Company
Detail Aging Report As Of 10/01/08 Based On
Invoice Date
Customers By ID From First To Last

Output Types
□ System Printer
☑ Portable Document File (.pdf)
□ XML Document (.xml)
□ Comma Separated Text (.csv)
□ Tab Delimited Text (.txt)

Invoice	Invoiced	PO Number	Order Number	Trm	Tp	Future	09/01/08 -09/30/08	08/02/08 -08/31/08	07/03/08 -08/01/08	06/03/08 -07/02/08	06/02/08
00-0100	Everest Industries										
	714.832.8113										
	Mr. James Rockville										
0005053	01/31/08										1,826.77
0005058	02/02/08										3,272.00
0005089	03/02/08										3,257.23
0005101	03/08/08										3,237.10
0005114	03/18/08										3,563.98
0005127	03/30/08										1,667.50
0009017	03/04/08	None	None	3	$						1,200.00
	Total for 00-0100 Everest Industries					0.00	0.00	0.00	0.00	0.00	18,024.58
00-0200	Western Sport Distributors										
	714.833.7100										
	Robert Mansfield										
0005054	01/31/08	None	None	CD	$						281.03
0005059	02/02/08	None	None	CD	$						1,304.51
0005090	03/02/08	None	None	CD	$						3,785.28
0005102	03/08/08	None	None	CD	$						2,580.29

Figure 6. The DocOut display window provides several choices for output type and fax or e-mail delivery method

sales orders, without requiring a third party solution. The Barista Fax/E-mail Queue does just that. The user can update this queue in a batch fashion to, for example, generate statements or update it interactively by browsing for and adding any previously archived document.

Barista Security

Many legacy applications fall short in their security features so the AddonSoftware developers decided to abandon the old security system and take advantage of Barista's role-based security. In Barista, developers can create users and security roles, and then assign one or more roles to each user; each security role provides application, form, and field-level security. Using the audit mechanism, developers can also log additions, deletions, or modifications to specified tables, a common requirement of the Sarbanes-Oxley Act.

Proof-of-the-pudding

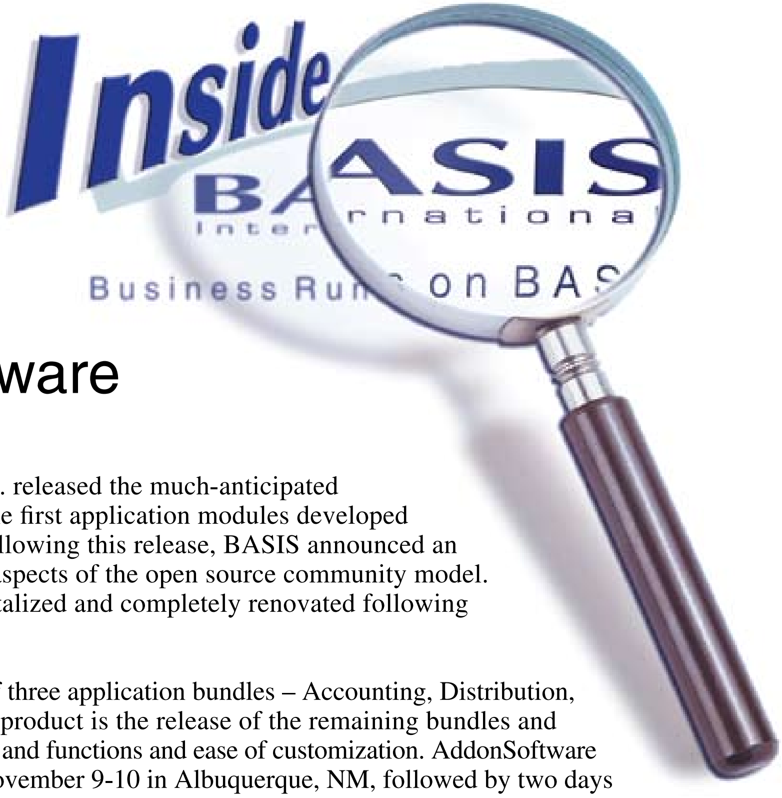
Thanks to the Barista Application Framework, AddonSoftware 8 is now available in a platform-independent graphical interface. Users will find that they can navigate through AddonSoftware forms in a way very similar to other graphical applications while appreciating the care taken to maintain keyboard-oriented, "heads-down" data entry functionality for data-centric forms. Barista has delivered on its productivity gains promise and then some!

The AddonSoftware reseller community is excited about the recent release of the AddonSoftware Accounting bundle that includes the Administrative, Accounts Payable, Accounts Receivable, and General Ledger components. Development is brewing away on the Distribution modules – Inventory, Purchase Orders, and Sales Orders – with Manufacturing and Payroll applications waiting to go into the frother.

Refer to the related Barista tutorials and documentation at
www.basis.com/products/devtools/barista/documentation

For more information about the AddonSoftware product or Partner Program, call 1.800.370.9131 or visit www.addonsoftware.com

Download AddonSoftware and Barista from www.basis.com/products/bbj/download.html
Choose Release Type "Current Release," select your platform, and check Optional Files "AddonSoftware 8.0" and "Barista." In the "Register for a BASIS License" window, select the "Requesting a 30-day evaluation license" checkbox if you are not already running a BASIS DVK License.

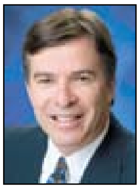


Barista Rejuvenates AddonSoftware

By Paul Yeomans

In August of 2008, BASIS International Ltd. released the much-anticipated AddonSoftware™ 8 Accounting Bundle, the first application modules developed with the Barista application framework. Following this release, BASIS announced an innovative partner program that embraces aspects of the open source community model. Trusted since 1981, AddonSoftware is revitalized and completely renovated following BASIS' June 2007 acquisition.

The cross-platform AddonSoftware 8 is comprised of three application bundles – Accounting, Distribution, and Manufacturing. The next strategic focus for the product is the release of the remaining bundles and re-training existing resellers on the many new features and functions and ease of customization. AddonSoftware will host a technical conference, AddCon2008, on November 9-10 in Albuquerque, NM, followed by two days of training. Packed with learning opportunities and demonstrations, conference sessions will explore topics that range from the migration process for the many users of earlier versions of AddonSoftware to Version 8, moving forward to new deployment options and customizing reports.



Paul Yeomans
Vertical Market
Account Manager

The combined power and efficiency of BBj, Barista, and the BASIS DBMS are on display with AddonSoftware 8. Using these tools, BASIS developers efficiently breathed new life into the accounting application and leapfrogged older technologies to deliver a modern cross-platform GUI application. Now vertical application developers can experience these same efficiencies for their own Barista applications.

Try AddonSoftware today. Download it from www.basis.com/downloads/bbj and select Optional Files "Barista" and "AddonSoftware."

Call 1.800.370.9131 US +1.505.338.4188 International for information about the innovative AddonSoftware Partner Program that also incorporates BASIS products.

AddonSoftware™

BASIS Improves b-commerce Support

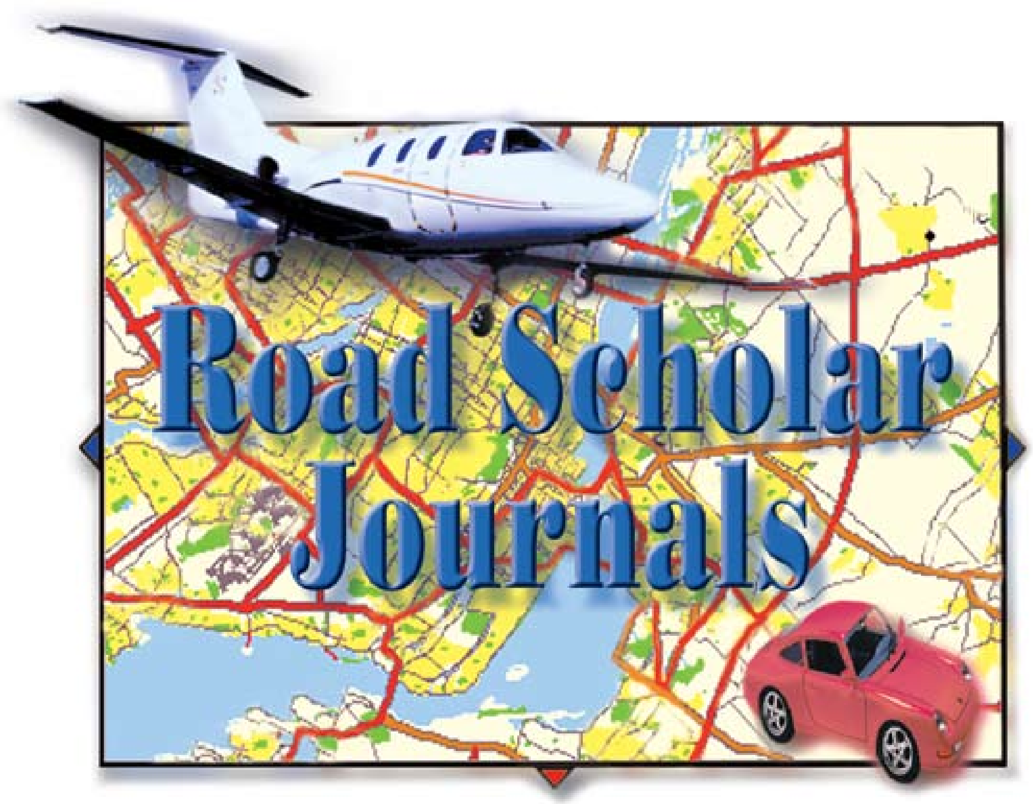
By Gale Robledo

BASIS International Ltd. continues to improve their support to b-commerce® users by giving them more functionality and information - making doing business with BASIS easy. The end user data entry tool in b-commerce allows BASIS partners to enter and link their customer information to the BASIS serial numbers. This linkage provides the ability to search by customer, run reports, and identify SAM renewal notices and invoices with customer information. Here is what one satisfied b-commerce user had to say…

> *"Having the end user data available in the b-commerce Web site is great new feature. This one extra step of entering our customers' information has saved me a significant amount of time placing orders or renewing SAM. Now that the customer information is displayed on BASIS e-mail notices and invoices, I no longer have to search through my serial number database to see which serial number belongs to which customer. This is a great tool and I recommend it highly."*
>
> **– Yvonne Mastrapa, ASCI of Miami Inc.**

b-commerce®

Gale Robledo
Account Manager



TechViews in MI and IL

As a rookie on the BASIS team - Vertical Market Account Manager for literally a few days - I joined Nico Spence on his March TechView tour to the Upper Midwest. There, he presented the 2008 BASIS Preferred Partner Program and the expanding functionality of the BBj®-based Barista® development tool. It was on-the-job-training in the greatest sense. I thoroughly enjoyed meeting our BASIS partners in Detroit, MI and Schaumberg, IL outside Chicago and hearing about their businesses, their successes, and their challenges.

The attendees at both TechViews enjoyed the presentations and were very enthusiastic about the power of Barista. They were particularly impressed with the financial impact of using the Barista Application Framework on application development and maintenance since its efficiencies dramatically reduce their customization and maintenance time. Time equals money. Several attendees shared their early success stories using the tool and the time it saved in delivering solutions to their customers. Also of interest, were the new PDF, fax, and e-mail output options in DocOut, and the Web services demonstration. Several partners requested copies of the demos to share with their organizations and customers and were pleased to learn that the demos are available as part of the product download. Additionally, their topics of interest can often be found in past BASIS International Advantage articles, available by searching on www.basis.com.

Needless to say, after this adventure I was even more excited about joining the BASIS team.

— by Paul Yeomans

OSAS Partners in Profit Conference

BASIS was honored once again to exhibit and present at the 2008 Open Systems Inc. "Partners in Profit" conference in Minneapolis, June 11-13, 2008. This annual event offers an extensive menu of breakout sessions on technical, partner, and product topics with 22 sessions alone in Sales and Marketing. Nico Spence and Dr. Kevin King presented the latest BASIS technology and the virtues of Barista in three successful breakout sessions –

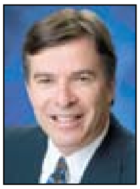
Enhancing OSAS with BASIS Technology Demonstrated how to use the language's database-centric features to enhance new and existing OSAS applications.

Effectively Utilizing the Newest BASIS Features with OSAS 7.x Presented the latest language features and how to enhance existing OSAS applications by incorporating them into the latest version of OSAS.

Barista – Fastest Development Tool in the West (and East) This presentation 'wowed' attendees with the data dictionary-based application development framework that easily triples development and maintenance productivity while delivering an up-to-date look and feel. The live demonstration took the OSAS data dictionary and within minutes added new GUI functionality to a familiar file maintenance task.

On Friday night of the conference, we enjoyed a special Mississippi River dinner cruise. As expected, the river ran high though it did not present the flooding problems felt downstream in Iowa and Illinois. We cruised under

continued...



Paul Yeomans
Vertical Market
Account Manager

Photo courtesy of Minnesota Department of Transportation

Interstate 35 bridge reconstruction

the enormous Interstate-35 bridge reconstruction - the $234M project that followed the tragic steel bridge collapse on August 1, 2007. The projected completion date was December 24, but the new concrete bridge actually opened on September 18 (must have used Barista to speed things along!). Under budget and more than three months ahead of schedule, contractors should earn close to $27 million in incentives by finishing early–food for thought for your next software development project with Barista! Hundreds have worked night and day and most holidays to complete this critical downtown route (note the proximity to Minneapolis in the photo above).

— by Paul Yeomans

Descore Showcase 2008

The final days of summer and first days of autumn mean it is time for the BASIS distributor in Canada, Descore Inc., to host their annual Descore Showcases. BASIS' newly appointed CEO, Nico Spence, and I traveled to Montreal and Toronto in mid-September to share all of the exciting product development news with our Canadian neighbors.

Nico speaks "BBx" in Montreal

Laurence Guiney
Senior Account Manager

Montreal is known as "the beautiful city" or La Belle Ville to the locals and with good reason - its European feel and marvelous architecture certainly did not disappoint. While French is the most common language in Montreal, those in attendance certainly understood that French was not the language that Nico came to speak. Nico was in Montreal, and later Toronto, to speak BBx®!

Nico's presentation, "How To Make BBx Work Smarter Not Harder for You" touched on many subjects including client-side objects, the ESQL file type, the latest version of the Enterprise Manager, and new controls such as sliders and spinners. Nico also covered the latest in BBx development tools such as a preview of the new AppBuilder and the application framework known as Barista®. Nico closed the session with a brief word on the upcoming AddonSoftware conference, AddCon2008, coming to Albuquerque, NM in November.

Toronto at night

All too quickly the trip came to an end, but as Nico and I left Canada, there was a sense of satisfaction that we had imparted a great deal of information on BBx that our customers will be able to use for years to come.

— by Laurence Guiney

BBj and Barista Make a Splash at CeBIT 2008

For the first time in several years, BASIS exhibited at the CeBIT tradeshow in Hannover, Germany. Nico Spence, CEO of BASIS International Ltd., and I, along with Andreas Timm and Angela Laermann from our European BASIS operations, teamed up in March to represent the BASIS products at the largest IT trade show in the world.

Our presentation of BBj® and Barista® fired up the imagination of most of the visitors at the BASIS booth.

continued...

The theme, "*Develop More Efficiently for the Java Platform with BBj and Barista,*" was an entirely new way to present BASIS products. We focused on BASIS' highly effective Java tools for any professional software developer; it was a great success. We established contact with more than a hundred new potential clients during detailed product presentations to IT professionals who experienced BASIS and Business BASIC for the first time.

Many of the visitors we spoke to reported their current Java projects slowed down after a while due to the very complicated toolsets that were required for maintaining and writing GUI applications. It was easy for them to see the immediate gains with BBj's very succinct, focused language and its three-tier architecture that, when combined with Barista, delivers rapid application development capabilities.

Project Consultants Angela Laermann and Andrea Timm at CeBIT 2008

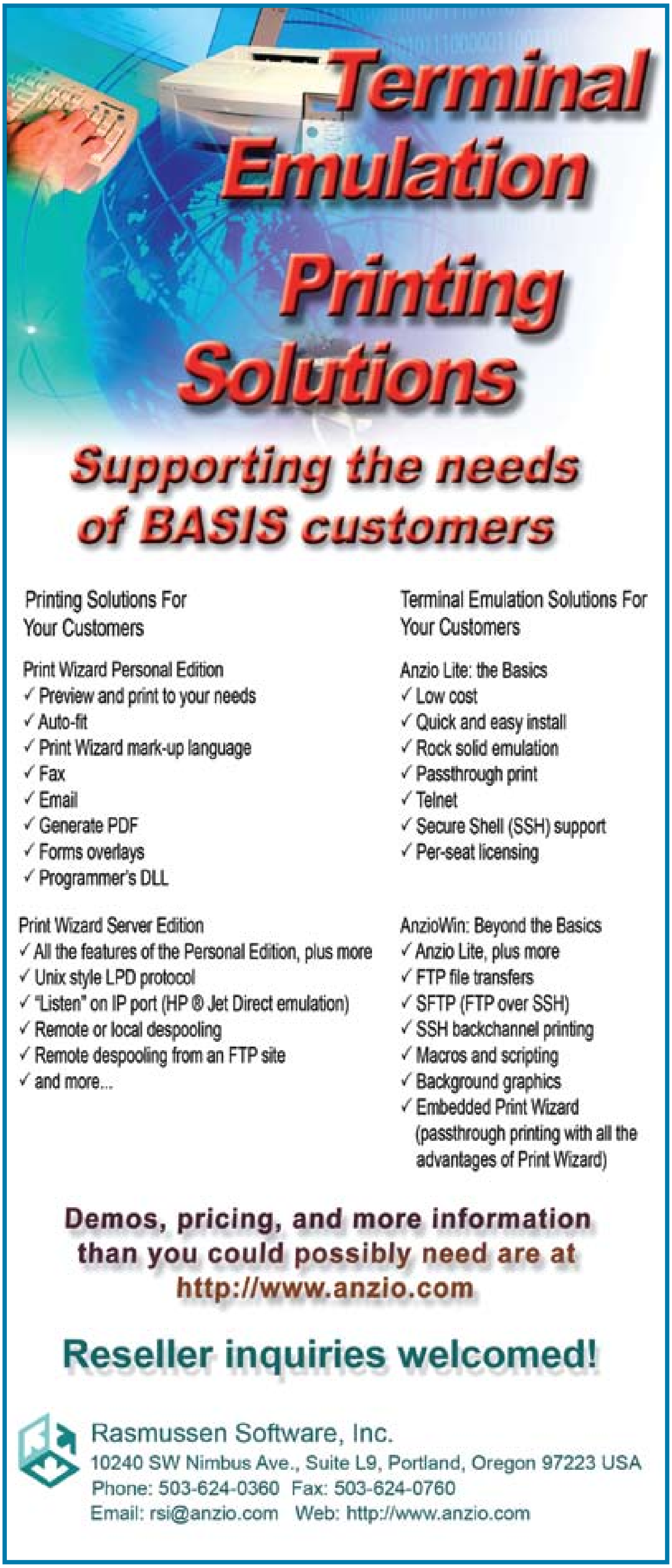
Since BBj integrates flawlessly with existing Java code, many Java developers have now begun to evaluate BBj, striving for productivity gains within their Java GUI projects.

As a result of the discussions with our CeBIT prospects, we are considering some valuable enhancement ideas and requests to the roadmap for both products. Coincidentally, BASIS engineers are already working on one suggestion; a Barista version capable of accessing standard SQL databases, making the product even more attractive and versatile to any software developer searching for rapid application development tools in the Java world.

Presenting our products at CeBIT 2008 confirms that BASIS and all BASIS customers are on the right track: BBj and Barista offer a wide variety of options to develop modern, powerful, and reliable business applications. Together, BBj and Barista leverage the benefits of each individual product. They deliver productivity gains and provide integration with Java code libraries while avoiding the pitfalls and hurdles of the large variety of complicated and disparate tools available in the market and community today.

Stephan Wald
Director Sales and
Technical Services

— by Stephan Wald



Who Ya Gonna Call?

By Janet Smith

BASIS International now offers many new and exciting products – Barista®, AddonSoftware™, and the BBx® family of products including BBj®. With so many new products and avenues of obtaining technical support – domestic, international, newsgroups – it can be confusing, to say the least. Here are some questions and guidelines for obtaining the most accurate and efficient response for the product in your location.

Ask yourself, *"Where did I purchase the product?"*

BASIS mainly supplies products and services through an international network channel of Distributors and Resellers. BASIS also supplies self-programming end-user corporations with product and services.

Application users should always contact their application developer first, be it an in-house development team or an authorized BASIS reseller. Most of the developers provide product support and are most familiar with the specifics of their application and configuration, far more so than the BASIS support team. Should you no longer have a relationship with your application provider, contact BASIS Sales for a list of authorized resellers in your geographic or vertical market area. International users should contact the international Distributor or Reseller in their region.

For Distributors, Resellers, and self-programming end user corporations who have purchased product directly from BASIS, use the following decision tree:

Ask yourself, *"What product am I using?"*

- **BBx-family product** (BBj, Visual PRO/5, PRO/5) –
  - Place a **phone** call to +1.505.345.5021, or
  - Send a message by **e-mail** to support@basis.com For support in Spanish, send an e-mail to soporte@basis.com, or
  - Submit an **online** BASIS e-Support Request Form at www.basis.com/support/support.html
- **Barista** – support is available via the Barista-List@basis.com newsgroup at www.basis.com/support/discussionforums.html
- **AddonSoftware** – support is available via the AddonSoftware@basis.com newsgroup. To join this newsgroup, refer to the Existing Partners page at www.addonsoftware.com
- **Evaluation license** – support is available by submitting an online BASIS e-Support Request for Evaluation and Emergency Licenses at www.basis.com/support/evalsupport.html
- **BBj development build** – support is available via the bbj-developer newsgroup at www.basis.com/support/discussionforums.html

Ask yourself, *"What information is required or most helpful to BASIS when assisting me?"*

BASIS Support Analysts require the following information to open an incident report:

- Product serial number(s)
- Name of the product
- Revision level
- Method of contacting you, including your name, your company, and a telephone or fax number or return e-mail address
- Detailed description of the problem, including any error messages and TCB(10) values. If possible, isolate the code block or function in which the error is occurring and provide us the values of any variables referenced in it.

In addition, the following information will also help the Support Analysts provide the best possible support and most accurate answers:

- What is the operating system and revision level?
- What version JRE are you using with BBj?
- Is the error reproducible or sporadic?
- Is the error isolated to a specific machine or user?
- Is the problem concerning a new installation or an existing system?
- Do you think the problem may be network related? If so, provide details about your network.

Janet Smith
Technical Support Supervisor

Summary

So, remember that your best port of call is always your application provider, be it an in-company MIS department or reseller organization. If you are a reseller organization that acquired BASIS product through a Distributor, then contact your Distributor for support who will involve BASIS support as they deem it necessary. For all those channel members who purchased their product from BASIS, we stand ready to provide the appropriate product support for your needs – we are waiting for your call or e-mail!

Have You Missed an Issue?

Find all of these articles and more on our Web site www.basis.com/advantage

Partnership

2007	#1	RCG Uses Marketing Skills to Achieve OSAS “Top Dog” Status
		Marex Returns Home to BASIS – A Personal Journey
2006	#1	Borealis Press Chooses Mac Platform
		BASIS Tours Europe With New Partnership Program
2005	#2	OSI Solidifies OSAS Investment with BBj
		QA Memos Web Service
2005	#1	BBx Generations – Our Progression in Time
		MARK SYSTEMS – A Revenue Stream Runs Through It
		Enhancing the Bouquet with BBj – KMK Software AG
2004	#1	Fit For the Future? – Midata Service GmbH
		BBj Sparks High Productivity at Heilind Electronics
2003	Q4	b’Using b-commerce
		BASIS Introduces Online Tutorials
		Maharam: BASIS Customer Profile
		Technical Resources on the BASIS Web Site
	Q2	SAM Plan – Managing Your Software Assets

Language / Interpreter

2007	#1	Spin, Slide, or View Your Data
		If You Have Choices, We Have Choosers
		BBj Spell Checker is all the Buzz
		A Tour of the BBjCharts API
		Leap from ‘Cut and Paste’ to ‘Drag and Drop’
		Desktop Data Delivered
		Catching the XML Wave
		A Deeper Voyage Into the BBjCharts API
		TRZ: Jars, Jars, and More Jars
2006	#1	A Primer for Using BBj Custom Objects
		Confessions of a Language Polygamist
		Applying Custom Objects to Existing Code
		The Scoop on 64-Bit Computing
		Visual PRO/5 6.0 with an XP or Vista L& F
		Type Checking With "bbjcpl"
2005	#2	Grid to Order
		Safe and Retrievable Intellectual Property
2005	#1	Input Validation: Veto Power
		Connection Pooling – Part 1
		Connection Pooling – Part 2
2004	#1	The Secrets of Secure Communication
		Why Use the BBjRecordSet?
2003	Q4	BBj Databound Control
		BBj Form Printing
		GUI Date Input With the BBj INPUTD Control
		Using the BBjRecordSet
	Q2	New Language Features Give Programmers More Choices
		Writing a Web Service in BBj
	Q1	BBj and Web Services
2002	Q2	Running BBj Code From Within Java; The JavaBBjBridge
	Q1	Performance Enhancements in BBj 2.0 - Print Batching
2001	Q2	From BBx to BBj
		The GUI Enhancements of BBj

Database Management System

2007	#1	Unleashing the Power of SPROCs Without SQL
		Solving the Data Warehousing Dilemma
2006	#1	Using Triggers to Maintain Database Integrity
		Using Stored Procedures to Add Business Logic to the Database
		ESQL Files: Constraining Your Data to Guarantee Integrity
2005	#2	PDF Now Also Means Perfectly Displayed Forms
		OpenOffice.org – a Sweet New Suite
		VKEYED Files – New Features for a New File Type
2005	#1	The BASIS DBMS – New 5.0 Features
2004	#1	Secure Your Future, Encrypt Your Data
		The Great Connection
2003	Q2	BASIS International Offers a Choice of Databases
		Capitalizing on BASIS’s Open Architecture
2002	Q1	Implementing Journaled Files and Transaction Tracking in BBj 2.0
		Route Data to Oracle & More with Plug-Ins
2001	Q2	Next Generation File Format: XKEYED Files

Development Tools

2007	#1	Watch the Form Gen Wizard Trans“form” Data
2006	#1	AppBuilder: The BASIS IDE Gets RAD GUI Development Integration
		Freedom of Choice: Using Object Code Completion in the IDE
		BASIS Puts Triggers to Work
2005	#2	Tuning the Performance Analyzer
		The Sate of the IDE
2005	#1	FormBuilder: BASIS IDE’s Better Cross-Platform Resource Builder
		TRZ: Logs, Logs, and More Logs
2004	#1	Inside BASIS: Software Product Version Control
2003	Q4	Managing Databases/Data Dictionaries with the BASIS IDE
		Viewing Data Files in the BASIS IDE
	Q2	Speeding Up the Development Process – The BASIS IDE
2002	Q3	Putting The BBj 2.0 Online Documentation to Work For You
		Using The Netbeans-Based BBjIDE to Develop/Debug (Part One)

System Administration

2006	#1	Solving the Locked Record ‘Whodunit’
		TRZ: The FAQ’s About 64-Bit BASIS Products
		Gain the Advantage – Use the Advantage Resource
2005	#2	Choices, Choices, Choices
		The Response File Responds Silently
		Safe and Retrievable Intellectual Property
		TRZ: Lock, Lock, Who’s Got the Lock?
2005	#1	New Install Puts You in the Driver’s Seat
		Taking Out The Garbage
2004	#1	Give Your Software a New Lease on Life
		Tech Resource Zone: Licenses and BLMs
		Without a Trace: The Silent Installation
2003	Q2	Enterprise Manager – Administration Made Easier
		Web Deployment Options With BASIS
2002	Q2	BBj Memory and Process Management
		BBj Thin Client Performance Tuning
2001	Q2	BBj: One License, One Install

Would you like to...

...add GUI bells and whistles that today's customers demand?

...take advantage of cool open source software or alternative operating systems?

...offer a modern interface and SQL access to your legacy data?

...be more productive by focusing on your vertical customization instead of common accounting modules?

...rejuvenate your revenue stream?

Then come and see what Barista and BBj did for AddonSoftware 8

rejuv-n-8 AddCon

November 9 -10

Register today at
www.addonsoftware.com

Come get rejuvenated!

AddonSoftware™