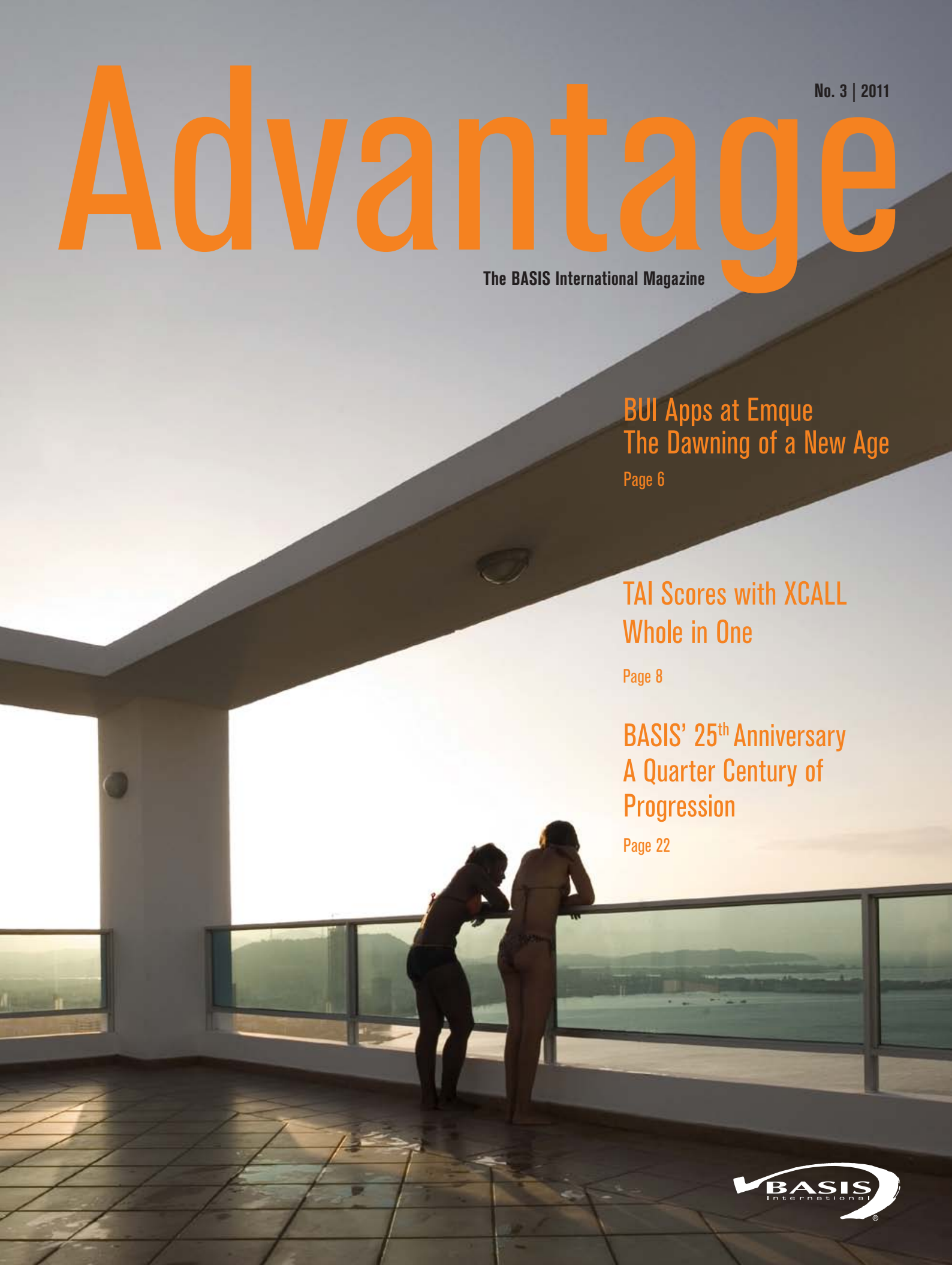
No. 3 | 2011

Advantage

The BASIS International Magazine



BUI Apps at Emque
The Dawning of a New Age

Page 6

TAI Scores with XCALL
Whole in One

Page 8

BASIS' 25th Anniversary
A Quarter Century of Progression

Page 22

BASIS International®

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.

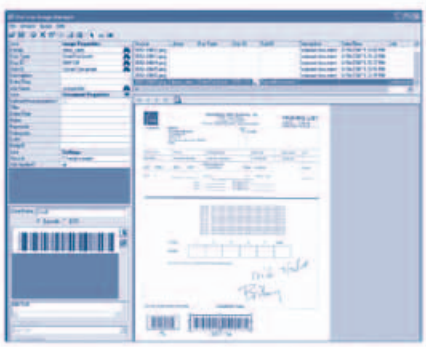
- Rules-based document archiving
- Archive concurrent with document printing
- Store multiple versions of a document
- 10 levels of user defined category indexing
- Document linking control
- Fast web browser-based retrieval
- Client API for application-based retrieval
- Index oriented archive browsing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

Universal web browser document retrieval



Document Image Manager

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

25 Years Later

Delivering an amazing return on your software investment!

In 1985, the monolithic hardware / software / application stack was stifling business productivity and entrepreneur's freedom of choice. The first release of the Business BASIC interpreter on a non-proprietary operating system occurred in the Fall of 1985 and signaled the fall from grace for the proprietary hardware vendors that heretofore controlled access to the hardware and operating system with an iron grip. Business software was set free.

Non-proprietary hardware and operating system combinations ushered in an era of openness, akin to the Wild West of yesteryear, that allowed developers to write software that would run on the platform of their choice. In the non-Business BASIC world, this irrational exuberance led developers to write code for the hottest platform of the day and when it fell out of favor they had to rewrite for the next hottest platform.

As this conundrum began to manifest itself, Java came along as a potential solution to this ever-changing world of OSs and even BASIS took advantage of Java to produce the most advanced platform independent GUI Business BASIC, BBj®, built on the Java platform. This seemed like a good solution for about a decade until the rise of smart phones, smart TVs, iPads, etc. that will never have a robust Java implementation on their virgin OSs.

Once again, developers world-wide found themselves with software written for proprietary systems that would require a rewrite if they ever hoped to compete on the new and most popular OSs. The last three decades in the life of non-Business BASIC application developers was all about chasing the latest/coolest platform and never quite catching it.

Fortunately, BASIS customers have missed much of that misery. Over the past 25 years, customers whose applications were written in Business BASIC did not have to change their application code to move from platform to platform, they just had to purchase an interpreter from BASIS to allow their software to run as well or better than it ever ran on the original platform for which it was written. Some of the most popular and notable platforms that have come along after many of the applications were originally written are Windows, Linux, and OSX.

However, new OSs are now being developed faster than JVMs and applications can be written to keep up with them. So, while the latest generation of non-BASIS software developers are recoding again for the Web browser using JavaScript (JS) and HTML, once again, BASIS customers are in a position to focus on business and not fret over this latest technology craze. Why? Because BASIS now interprets BBj code to either run in a JVM or in a JS-enabled Web browser!

Since BASIS now offers support for the latest Browser User Interface (BUI for short), customers can again rest assured that their applications will still run on the latest exotic OSs that come on handheld devices, tablets, and televisions. Because they all come Internet-connected, the exotic OSs have one thing in common with the traditional OSs – JS-enabled browsers. Any GUI application that was written for Windows in Visual PRO/5® will now run on any of the GUI-enabled Java platforms via BBj as well as on any JS-enabled browser, regardless of the underlying OS.

The past 25 years have passed in a blur as we kept up with the constantly changing technical environment, shielding you from the need to rewrite your applications for technology's sake. While we cannot begin to imagine what the next twenty-five years will bring, we are confident that we will be able to continue our tradition of delivering cross-platform compatibility to you and the community at large, no matter how fast of a pace the technical giants like Google, Microsoft, Linux, Apple, Oracle, etc. might set.

Whether your software is CUI, GUI, or BUI, the BASIS commitment still stands strong … to make your software investments deliver a continuous return well beyond their original break-even point.

Nico Spence
Chairman & CEO

Dr. Kevin W. King
President & CIO

Peggy Lewis
COO

Partnership

Language / Interpreter

DBMS

Development Tools

▸▸▸ Article continues online

System Administration

Building Blocks

Columns

By **Stephan Wald**
Director Sales and Technical Services

Lest darkness fall …

When I was writing the editorial for the last issue of our Advantage magazine ("I love this company …"), I already had this strong feeling that the new single source capabilities of BBj of writing real three tier Java Swing GUI applications and Javascript-Ajax applications with the same source code would be a cool feature. Since then, there had been many occasions that gave me confirmation.

The IT industry probably always needs its own villains: While IBM and Microsoft had filled out this role for a long while, Larry Ellison assumed the role of the leading sod of the industry.

Would we just be offering one of these frameworks, IDEs and RAD tools for the Java developer, I definitely would be concerned about the current governance of Oracle over Sun, Solaris, OpenOffice and Java. But the history of our company has shown that BASIS did always go a path allowing us to achieve a very solid growth, for more than 25 years now.

Our long term strategy – offering a fully-fledged toolset for the development of state-of-the-art business applications while always keeping upwards-compatibility in mind - allows our customers to even run 15 year old GUI code in today's browsers and on any current mobile devices. See the article about EMQUE in this issue.

So there will be again good news for our customers: No matter how Oracle and Google end up, no matter if Java FX will make it to more mobile devices or not, no mater if IE will prevail or Chrome will win, just go ahead as you always did, focus on business logic, write your code in BBj and Barista, and lean back. BASIS protected your investment in code and knowledge for 25 years until today, and so will we do for yet another 25 years to come.

"Lest Darkness Fall" is an alternate history science fiction novel written in 1939 by author L. Sprague de Camp. I'm looking forward to the first IT-industry related novel living upon the mind-game of "what if things in history would have taken a slightly different twist". ■

http://en.wikipedia.org/wiki/Lest_Darkness_Fall

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

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

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

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

www.basis.com/company/legal/trademarks.html

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

Subscribe at www.basis.com/advantage/subs.html

BASIS Europe Distribution GmbH
Nell-Breuning-Allee 6
66115 Saarbrücken, Germany

Phone: +49 681 96814-0
Fax: +49 681 96814-33

www.basis-europe.eu
info@basis-europe.eu

By **Susan Darling**
Technical / Marketing Writer

The Dawning of a New Age with BUI Apps at EMQUE

"This stuff really works … it's not just cool-looking!"

For nearly 10 years, EMQUE Consultants, Inc., a developer of a custom fully-integrated suite of Visual PRO/5® powered applications for the commercial construction industry, dreamed about running their application on a reliable and easy-to-use mobile device. The timing just wasn't right.

Then, earlier this year, it happened. All the stars aligned. First, Apple rocked the world with the release of their iPad, the first widely popular tablet computer; then BASIS blew open the development doors with their Browser User Interface (BUI). EMQUE Consultants Inc. finally reached the point where they could give their application a fresh look.

The Dream Comes True

"We've seen all the planets align in the harmonic convergence. Perhaps that was the age of Aquarius, but this technological convergence is clearly the age of the BUI app!" says Mike Quagliarello, President and Chief Architect of EMQUE Consultants, Inc. "When I read the BASIS Marketing Communiqué announcing BUI, we paid close attention to the 'running in the browser' feature and just waited in the wings for BASIS to release 10.0 so we could get our hands on it. We dove right into the nightly build and now keep up with the daily BASIS developments and fixes."

EMQUE's first suite of programs, "The Foreman," announced on September 21, 2010, is an add-on application to their existing software solutions. In the 45 days since that announcement (tinyurl.com/3y88dme), EMQUE has three installations, with another two scheduled in the coming weeks. Using this app, a job foreman can write quotes, place orders, check inventory, enter payroll data, etc. on the spot, directly from the field, for immediate uploading to the main office. The Foreman is more than just a pretty face, more than an easy button; it addresses the paperwork monster that is a nightmare to control in the construction industry.

Using the Dragon Dictation (www.dragonmobileapps.com) "turns talk into text" application, job foremen can opt to work hands free by just talking into the device and/or copying and pasting dictated text right into the app. "And my hero, Jim Douglas, wrote a signature capture feature that makes it jaw dropping when people see it. Not only can people 'talk' to their iPad into a work ticket, but customers can sign the ticket on the iPad and immediately e-mail it back to the customer, all automagically on iPad!" Quagliarello continues, "This stuff really works … it's not just cool-looking!" Future applications will run on smart phones such as Android and iPhone as well as iPad-type tablets.

A Look Back

Quagliarello explained that their mobile device focus was on iPad and heeded early advice from Dr. Kevin King, BASIS President and CIO, to process as much as possible on the server and then push to the device. "To get to the final working application, we had to rethink our Visual PRO/5 architectured application model. And working in a new environment along with BBj® had some challenges. The thing that really kept us going was the support from BASIS. Their engineers, particularly Adam Hawthorne and Jim Douglas, worked closely with us to modify our module for iPad. Mike Hack, our software engineer, and I were grateful for their advice and guidance."

As EMQUE pioneered the first conversion to BUI, there are some tips they learned that might be of value to the next group of developers to follow in their footsteps. Grid representation has a ways to go on iPad so they avoided grids wherever possible. Also, iPad doesn't support listboxes yet so EMQUE reworked their app to use listbuttons. Quagliarello explained, "Jim Douglas did something really special with the listbutton. He made it so fast that even if it contains 20K items, the whole screen loads in 20 seconds! No back and forth to/from the server. I don't understand how he did it but it works! Again, Jim is my hero." And as mentioned earlier, it is most efficient to process as much of the app as possible on the server first, then push to the device to avoid going back and forth between the client and server.

A Look Beyond

To compliment The Foreman, EMQUE also created a similar custom BUI app called "The Owner." This suite, designed for an owner, provides a snap shot of cash flow, receivables, aging, customers and sends statements via e-mail. Quagliarello shared that his first 'owner' installation has been busy running around Manhattan visiting all his job sites with iPad in hand, making great strides in his iPad proficiency. The customer said that he can normally fumble through the app in the office but EMQUE made it simplistic for him or anyone to run on a tablet. In fact, this customer had just sent an e-mail from London saying 'ya, know, I've been checking on my jobs and noticed that 3G is really a lot faster here in England than in NY!' Have iPad, will travel … and owners can keep tabs on their companies from any corner of the world.

A Look Ahead

From an application standpoint, EMQUE wants to run their entire application in the browser with the look of a Web page, yet still function the way clients expect. "We still have things to do first, and iPad and BUI aren't quite ready yet either, but hopefully we can move in that direction next year." From a marketing standpoint, Quagliarello is positioning a few very key clients to use the new technology and leveraging them to entice other customers and eventually new clients. "Mobile devices, especially iPad, are culturally ahead of their time for many of my customers. That is the biggest challenge but soon that will change. I see potential for springboarding into iPad applications outside of our current construction market … the whole world has opened up to us now."

In these economic times, EMQUE, and perhaps their own customers, continue to wisely evaluate the vendors with whom they choose to do business. Considerations such as "is this the best vendor to partner with?" and "will this company still be around in 5 years?" are valuable in analyzing current business relationships. EMQUE is certain this move to BUI has, in fact, secured their place in the market and extended confidence to their customers of EMQUE'S own future. Quagliarello explains, "we have squashed any concerns customers might have had about EMQUE flattening out or dying in the future. We have stunned a number of people with what we are able to do with BUI." ■

EMQUE

EMQUE Consultants, Inc. develops and custom designs, for commercial construction contractors, a fully integrated suite of applications employing best industry practices. Perfect Project, EMQUE's name for the created systems, highlights its seamless integration into an office automation suite. A Perfect Project installation makes financial analysis of business operations a breeze, helping customers know when they are profitable and empowering them to make better financial decisions.

Mike Quagliarello, founder of EMQUE, is President and Chief Architect.

Visit EMQUE at www.emque.com.

By **Susan Darling**
Technical / Marketing Writer

TAI Scores a Whole in One With XCALL

"To add a one-line XCALL that runs BBj programs from within the standard legacy Visual PRO/5 programs … it is absolutely huge!"

Great success at TAI Club Management Systems comes, in part, to their commitment to meeting their customers' needs. Recently, TAI met the challenge of evolving e-mail security by enhancing their e-mailing capabilities from within their Membership Management System, specifically while sending member statements via e-mail.

Michael Ley, Senior Developer explained, "We had a called program that communicated with our e-mail server but some password verification issues prevented us from using that program. This situation forced us to look for other options." President Mike Talbot adds, "When I saw the announcement about XCALL coming out, I got really excited. To add a one-line XCALL that runs BBj® programs from within the standard legacy Visual PRO/5® programs...it is absolutely huge!" Since this module was written in Visual PRO/5, their obvious option was to use BBj's e-mailing capability by way of the XCALL Server.

BASIS was the Caddie

Talbot and Ley were one of the first BASIS customers to take a swing at this new implementation. As seasoned veterans playing on this virgin XCALL fairway, they did hit a few bunkers and other obstacles to overcome on opening day. On par with their longstanding commitment to customer satisfaction, BASIS engineers worked diligently with TAI to clear the course. "The BASIS engineering team was a tremendous asset in working with us to figure out those little divots and add the fixes into the latest release. Now it is as advertised and what BASIS says is accurate and true, we really did only have to change one line of code in Visual PRO/5 to run the XCALL. And it ran fabulously!" Talbot explained.

XCALL Bridges the Visual-PRO/5-to-BBj Gap

"Speaking for all vendors with legacy GUI code, at the outset it looks like a transition to BBj is going to be smooth and great and tremendous, but when reality sets in, we see that it really is going to take some time," Talbot explains.

One wants to enhance the application with the power of BBj as part of the transition. The plethora of new functionality and language capabilities such as the new object-oriented syntax and callback event handling paradigm are difficult to resist but can necessitate significant code changes.

Talbot continues, "XCALL allows a more incremental transition by harnessing the power of BBj in certain utilities and common programs and then back-filling them into the legacy code so there aren't two full sections of code doing the same thing. I only have to maintain one code line and use XCALL to back fill it into the old legacy stuff. It's a tremendous tool to go forward...to bridge the gap to get from traditional Visual PRO/5 programming to BBj. Now that we understand the strength of XCALL it actually is causing us to hold back a bit to think about how we should now attack the conversion process. While it's opened up a lot of doors, we now need to go back and look at the blueprints to make sure we have the doors lined up properly."

Feedback from the Gallery

"End users are seeing great benefit from our conversions to BBj, although they aren't aware that BBj is responsible. They just see product that works like they expect it to."

Ready for the BBj Course?

"I drug my feet until they showed me the pretty graphs at TechCon. I figured then that BBj was stable and mature enough to use as a development platform. Since we started moving over to it, although the process to relearn a new way to doing things is always a challenge, BBj turned out to be much easier than I, a non-Java programmer, originally anticipated. Logic is logic and once I got the hang of it and the flow, things made sense. The great thing about it is that for someone like me who doesn't have the Java experience, I could still become very adapt at Java programming. Those programmers experienced with Java have much more capability that they can bring into the product. So from a BASIS standpoint, it was a tremendous move to go in that direction (Java) and the faster that TAI is able to get there, the better off our product offering is going to be to our industry."

Summary

What's next for TAI? "All our current new development is done in BBj and we're just picking the modules and systems to convert over. We started with our banquet module then small front desk check-ins for health clubs. Right now we are evaluating whether to redo membership or start payroll or revamp another module," responds Talbot.

"For us, it's the little things that BBj offers, like grid management. It doesn't have the overhead and bulkiness the way grids used to be; having it be so much more simple changes the development time, which is why I can't stand to develop any more in Visual PRO/5. I want to move ahead with BBj for everything, as quickly as we possibly can."

So follow TAI's lead. Add the XCALL club to your bag and score a "whole" in one! ■

TAI Club Management Systems provides solutions to the resort, private club, and public golf course market sectors. Their foundation is the TAI Club Management System software suite, the most continuously installed club software system in the United States. TAI Club Management Systems...small enough to care, large enough to achieve your goals!

Mike Talbot is President of TAI Club Management Systems.

Visit TAI Club Management Systems at www.taiconsulting.com.

By **Dr. Kevin W. King**
President & CIO

Our Salvation is in the Cloud

"Moving to the Cloud we manage to reduce the total cost of ownership for ourselves and our customers"

This sounds like the title of a sermon augmented by cherubim and seraphim, but it is really the title of a message designed to provide customers with the collateral and confidence necessary to move their clients into the cloud when and where it makes good business sense to do so.

The Problem

BASIS supplies product and licenses to customers and countries all over the world that depend on our servers functioning 7 x 24. The robustness of these servers have a direct correlation with the confidence and comfort that the customers have with their suppliers. We certainly could spend a lot of money bringing redundancy and robustness to our organization without moving to cloud-based computing. The requirements to achieve a 99.99% Service Level Agreement (SLA) include, but are not limited to, multiple Internet providers, backup power generators, multiple load-balanced servers, off-site backup and restore facilities, multiple cross-trained engineers to support the infrastructure, etc. Even after investing all of the money necessary to attain this level of redundancy, a thief, a fire, or a terrorist could still render BASIS non-functional and non-performing with very little effort and even less imagination.

The Solution

By moving all our business processes into the cloud (warehouses full of servers somewhere in the world), many of the requisite redundancies became automatic. Once in the cloud, we gained thousands of servers, multiple power sources, multiple Internet access points, replicated databases, automated backups to numerous data centers, e-mail and document warehousing, and ubiquitous VoIP service. Moving to the cloud eliminated the need to size the project servers because we set the systems up on micro-sized virtual machines that cost only a penny or two per hour to run.

Once the servers went live, we selected larger and larger virtual machines until the servers performed at acceptable levels. As the size of the demand grows or shrinks, we can restart the virtual machines at the new size in approximately two minutes. This ability eliminates the guesswork involved in most server purchasing discussions. There are no up front costs for any of the servers and it is pay-as-we-go that eliminates potential worry about outgrowing an expensive in-house server before achieving its five-year payback window.

The Technology

BASIS chose Google Apps as the provider of cloud-based e-mail, documents, intranets, extranets, messaging, and telephony services. As of late 2010, Google provides these services to BASIS and over 3,000,000 other businesses, which gives us a great sense of confidence and a contractually guaranteed SLA of 99.99%.

Next, we chose Amazon Web Services (AWS) as the provider of cloud-based servers. In 2010, Amazon began their fifth year offering servers to the public. This is a strong history for cloud-based computing and, because AWS has warehouses on the east coast and west coast of the United States, Asia, and Europe, they were the best fit. Furthermore, since we distribute products in countries all around the world, it was important to choose a world wide cloud provider as well.

As of the date of the writing of this article, BASIS has servers running on the east and west coasts of the US powering numerous Web servers; www.basis.com, www.poweredbybbj.com, www.addonsoftware.com, bugs.basis.com, etc. Additionally, all of the product downloads utilize Amazon's Cloud Front technology, so the Amazon data center that is the closest in proximity to the downloader/customer pushes the files.

Downloads occur in such a completely automated and transparent way that end users have no idea which data center they are downloading from and yet, benefit from the fastest possible download of all BASIS products.

In addition, we moved all BASIS accounting and order fulfillment, and other business processes, to the cloud, as well as transferred the BBj®, Barista®, and AddonSoftware® product building processes out of the building. Furthermore, the cloud hosts the 32- and 64-bit servers used to test the products on Windows servers, half a dozen flavors of 32- and 64-bit Linux, and Solaris on Intel, all done automatically using the Hudson open source extensible continuous integration server software.

The Hudson software detects the changes as they are checked into the SVN source code archive and automatically starts slave machines in the cloud to build, package, test, and publish the development builds without any human intervention.

Most of the machines used in this process are only needed for a few hours per week so we can build and test products as quickly as possible, without upfront cost of the large hardware installations or the maintenance costs associated with maintaining large server farms. The cloud warehouse has tens of thousands of servers always available to meet our ever changing demands.

The Products

One would expect organizations like Google and Amazon to have cloud-based product suites. It might be a little bit surprising that all current BASIS products are also cloud-enabled. With the release of the BASIS Product Suite version 10.0x, all products work in the cloud, thanks to an enhancement to the BASIS License Manager.

To run Visual PRO/5®, PRO/5®, BBj, Barista, AddonSoftware, AND your favorite BASIS application in the cloud, just contact BASIS Sales and let them know that you are moving your license into the cloud so they can make the necessary changes to your serial number. Your users will thank you for the cost savings, redundancy, robustness, and simplicity of the transition from the server room in their building, to their server room in the cloud.

Summary

BASIS moved to the cloud to bolster our Business Continuation/Disaster Recovery plan. During the process, we learned what needed to be done to make our own products cloud-capable and how to reduce the total cost of ownership for ourselves and our customers as we embraced the mature and popular 21st century technology, Cloud Computing.

Is it any wonder that our salvation is in the cloud? Perhaps yours is too. ■

- To experience BASIS Products running in the cloud, go to www.poweredbybbj.com and see the following BASIS functionalities:
  - The Web page is running on a cloud server
  - BUI b-comm uses BASIS BUI technology from the cloud
  - GUI b-comm uses BASIS BBj thin client technology from the cloud
  - The accounting application under the BUI and GUI b-comm interfaces are powered by 20-year old BBx® code in the cloud
  - Downloads are facilitated by Cloudfront for optimal performance, and
  - The https://bugzilla.basis.com and www.addonsoftware.com sites are on Web servers running in the cloud accessing multiple MYSQL databases running on an Amazon RDS instance in the cloud
- For up-to-date information
  - on Amazon Web Services, go to aws.amazon.com
  - on BASIS Technology go to www.basis.com
  - on Google Apps, go to www.google.com/a
  - on Hudson, go to hudson-ci.org
  - on SVN, go to subversion.apache.org

By **Mike Phelps**
Software Programmer

BBj IDE in the Beginning – Then and Now

In the beginning, Man created the command line editor. The editor was without form, and void; and darkness was on the face of the Business BASIC console. Man's fingers hovered over the face of the console, typing things like ed 1020c[early]r[stone-age].

Then Man said "Let there be full-screen editing" and there was _edit. And Man saw the text-based full screen editor, that it still wasn't good enough.

Then Man said "Let there be graphical user interfaces, with windows and mice and integration."

And Man gathered together a graphical editor and a form designer and a compiler and a debugger and other previously disconnected developer tools, and Man called the result an Integrated Development Environment. And Man saw that it was good.

Man is not yet finished with creation, of course. At BASIS we are constantly engaged in making improvements to our own Integrated Development Environment. Even though perfection is a long way off, we'd like to show you the latest iteration of the BASIS IDE built on NetBeans from Sun Microsystems (now Oracle) and extended by BASIS with plug-in modules to give it the capability of developing Business BASIC applications.

This e-article highlights some of its most helpful features, discusses what it takes to get it running, and reveals some insider tips. If you have already tried the IDE, you may pick up some useful information you hadn't heard before. If you haven't tried the IDE, you may find the time has come to give it a try!

Part I - Configuring the IDE for Your Development Project

To install the BASIS IDE, you need a Java 1.6 JDK and the latest version of BBj® (as of BBj 10, there is no longer any need to run the IDE with a Java 1.5 JDK).

Running the IDE with only a JRE (Java Runtime Environment) instead of a full JDK is possible but not a good idea, since you will be missing the Java debugging tools. The IDE will complain loudly about this condition. Likewise, it is not a good idea to attempt to use the IDE component from an older version of BBj with an installation of a newer version of BBj, or vice-versa. The BASIS-designed plug-in modulesthat make the IDE Business BASIC-capable are closely tied to the specific version of BBj they are released with. Mixing and matching different versions of the IDE and BBj will result in subtle bugs or even total failure of various features.

We are sometimes asked "can I install the BASIS IDE on a central server and run in a multi-user environment, eliminating the need to install it on individual developer's machines?"

The answer is a definite maybe. Nothing in NetBeans prevents it outright, but the BASIS installation and plug-in modules configuration were not designed with this in mind. ▸ ▸ ▸

Continued at
links.basis.com/ide

By **Shaun Haney**
Quality Assurance Engineer

If it's Easier in BBj, Then Just XCALL it Forward

Ralph, a software engineer for Fictitious Enterprises, was perplexed.

Until now, Ralph's in-house Enterprise Resource Planning (ERP) application could do everything the company needed, but then the day came when the sales force asked for a pie chart showing the clients that accounted for the majority of sales by telephone. Making a pie chart is possible but not terribly easy using drawing mnemonics in (V)PRO/5 (Visual PRO/5® and PRO/5®). He knew that there was a BBj control that was a faster and more attractive way to make a pie chart but the ERP application, written in (V)PRO/5 for CUI or GUI, had never been run in BBj®. Additionally, BBj's charts are full-featured and offer built-in functionality that allows users to interact with them and dynamically change their contents. By right-clicking and selecting options from the popup menu, users can zoom in and out, modify the axis ranges, and even save out a local copy of the chart as an image on their computer. Ralph knew he would eventually move the whole ERP application to BBj and that the transition would be seamless but on this particular day, he just needed a convenient way to create a compelling pie chart.

Was there an easy solution? Absolutely. Ralph remembered attending a Java Break with BASIS and learning that (V)PRO/5 version 10.00 offers the new XCALL verb!

Introducing the XCALL Verb

While the XCALL syntax is similar to the CALL verb, XCALL allows an earlier generation of BBx® program to call a BBj program or subroutine. The called program or subroutine in BBj is exactly as one would write it in (V)PRO/5: Create a label if desired, create an ENTER statement with the needed parameters, write whatever code is needed to perform the task, and then end the subroutine or program with an EXIT statement.

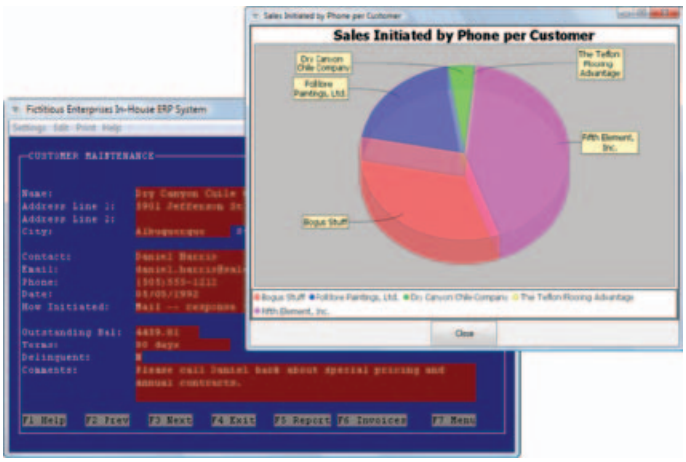


Figure 1. Sample pie chart from within the ERP application

This new verb allows (V)PRO/5 to call BBj and run any code in the program or subroutine, and even retrieve results of the call through variables that were passed by reference into the call. XCALL would perfectly fulfill the sale force's request by allowing the ERP program to tell BBj to display a pie chart with a specified set of values all in one elegant line!

Ralph decided that instead of spending days trying to create a reasonably attractive pie chart himself, he would simply take advantage of the existing charting functionality provided in BBj via XCALL!

Follow along online with Ralph as he implements and deploys a pie chart (Figure 1) on his end users' desktops from within his company's ERP application, using XCALL. ▸ ▸ ▸

Continued at
links.basis.com/xcall

By **Nick Decker**
Engineering Supervisor

Battle of the Browsers – BUI Wins

The World Wide Web has greatly matured over the last couple of decades and has now become an indispensable and integral part of our lives. While a browser's main job still is to deliver information to a user, the most radical change lately has been the aspect of interactivity. In the old days, Web pages presented static information with a few hyperlinks sprinkled throughout. In contrast, today's Web pages take advantage of several new technologies to present dynamic content. We can now view a Web site that customized all of its content – even advertisements! – to the visitor and allows the users to interact with the site to accomplish their specific goals, rather than being limited to passively viewing predefined content that may not apply. New technologies such as AJAX play a huge role in this change, allowing webmasters to design Web sites that change page content on-the-fly without having to redirect the user to a new page. The end result is a rich, immersive, and topical Web experience that customers have come to expect. BASIS' Browser User Interface (BUI) dovetails into this user-interactive experience perfectly, by running fully-functional Business BASIC applications natively in the client's Web browser.

A Browser Isn't a Browser, Isn't a Browser

Over the years, one of the biggest challenges for webmasters was ensuring that their Web sites looked good and performed well on a variety of different browsers. Although Web standards exist for the underlying technologies such as HTML and CSS specifications, it is well known that some browsers did not fully implement the specs, had bugs that prevented features from working properly, or in some cases, purposely eschewed proper behavior in favor of supporting older broken behavior for backwards compatibility. These challenges are trying at best for your typical webmaster, and as time goes on things haven't improved as much as one would think or hope.

On the the bright side, however, Microsoft's Internet Explorer 8 (IE8) serves as a example of how browser manufacturers are now striving to adhere to industry standards. As a case in point, IE8 offers the ability to render a page in 'strict mode' that boasts stringent adherence to W3C Web standards (in addition to the default 'quirks mode' that favors backwards compatibility with older versions of IE). This is good news for Web developers, as it eases the burden of development, but it is also telling in that it shows that browsers must now work well in order to remain competitive. The olden days of using the default browser installed on a system are over, as stand-alone browsers such as Mozilla's Firefox and Google's Chrome have become extraordinarily successful. Their increased development schedules, improved security mechanisms, advanced plug-in and extension architectures, and improved reliability and performance allowed them to leap-frog the competition from Redmond and have sparked a heated competition for the user's desktop and handheld devices.

The other bit of good news is that despite the vast number of differences in browsers today, most of the onus of making BUI programs run seamlessly on a variety of browsers has been taken on by the GWT (Google Web Toolkit) team and the BASIS engineers. Even though differences do exist from browser to browser, in many cases the BBx developer is shielded from these anomalies as the same application code runs similarly in multiple browsers. There still are, and always will be, numerous differences between browsers but many of these differences are benign and workarounds exist for some of the more egregious ones.

Browser Differences for BUI Applications

In a nutshell, a Web browser's task is to take building blocks such as HTML code, CSS styles, text, fonts, images, and scripts such as JavaScript and put them all together to present a coherent user interface. Given the complexities of the source combined with adherence to various levels of specifications, it's easy to see how each browser's rendering engine could come up with a slightly different result. In some cases, the differences are subtle such as a control

looking slightly different or sized differently in one browser compared to another. In other cases, the differences may be profound as various JavaScript speed tests have shown some browsers to be over 1,000% slower than their competitors in certain tests. All of these variations stem from the fact that browsers use proprietary engines to perform complex tasks such as layout and script execution.

As time goes by, browsers get more and more competitive with one another, vying for the title of fastest browser. Manufacturers realize that a fast browser results in a speedy, smoother, more satisfying Web experience. JavaScript performance in particular is highly contested, as it is ubiquitous and is responsible for critical concepts such as client-side validation, document manipulation, animation, and more. JavaScript speed is also one of the key factors that determines how quickly and responsively a BUI application performs, making it an important consideration when choosing a target browser to deploy a BUI application suite.

Look and Feel Considerations

As mentioned earlier, various controls may render slightly differently across various browsers. Often these differences are so minimal that it's not likely anyone would ever notice. However, differences become more prevalent when the browser changes the look of a particular control, such as a button, to adhere to the standard for that browser on the target operating system. To illustrate this point, take a closer look at the BBjListButton control that appears in the BUI Customer Maintenance demo. Figure 1 also illustrates these; the shape and coloring of the control, the look of the drop down box on the right side of the control, and even the font and text alignment.

Figure 1. The BBjListButton in different browsers and platforms

In addition to the 'Look' portion, the 'Feel' of the controls varies as well. Using the same BBjListButton control example, selecting the drop down button on the right of the control (usually denoted by a disclosure triangle) causes the control to present the user with a list of possible choices. The type of list presented will differ, with some resulting in a drop down list, some with a pop up list (see Figure 2), and a native picker control on Mobile Safari.

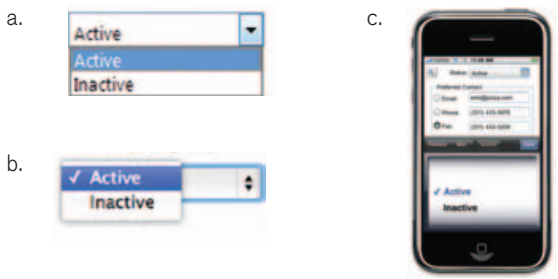


Figure 2. Drop down and pop up list for the BBjListButton on a. IE 8, b. Chrome, and c. Mobile Safari

Speed Considerations

Now that browsers play such a large role in our everyday computing life, browser manufacturers know that speed makes a huge difference in how well the market receives a browser and ultimately its popularity. Evidence of this comes from several sources, including the manufacturers themselves, who prominently display statistics as selling points like Opera's "Our further-optimized JavaScript engine is over 50% faster than in Opera 10.5" (and that's just one of the many differences in a 'point release' going from 10.5 to 10.6!). As manufacturers vie for the title of fastest browser on the planet, the big winner in the browser war are the end users. They, after all, get to enjoy the rewards that are a direct result of this heated competition. Over the years, browsers have become several times faster due to this browser race and have been further spurred on by popular benchmark suites like SunSpider. Several different benchmarking suites exist, making it relatively easy to make direct JavaScript speed comparisons between multiple browsers. This information has proven interesting when comparing alpha and beta versions of upcoming browsers, but is extremely valuable and directly applicable when reviewing the current crop of released browsers in order to provide a recommendation for your department or end users.

One of the more challenging problems facing today's system administrators is that a browser like Microsoft's Internet Explorer is still the most popular, despite the fact that it's one of the slowest browsers available. According to StatCounter (see Figure 3), IE still holds the lion's share of the global browser market. However, it's popularity has been steadily dwindling over the years going from over 70% a few years ago to dipping below 50% for the first time in September of 2010. This is proof that newcomers, such as the super-fast Chrome,

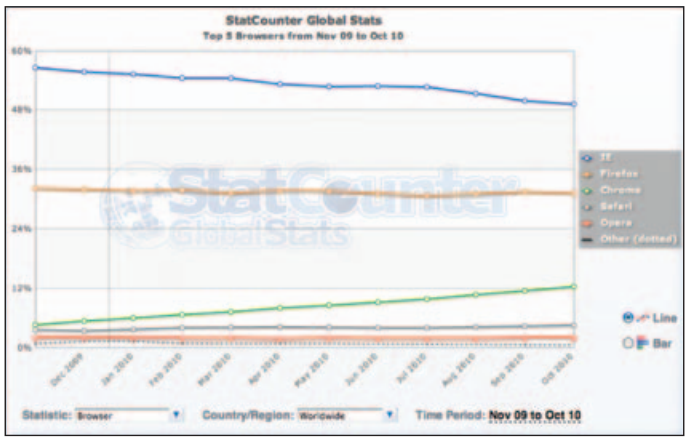


Figure 3. StatCounter's global browser share for the last year

have been steadily gaining acceptance over the last year, slowly eating away at IE's dominance. The writing is on the wall for the older, slower browsers – improve or be left behind.

To Microsoft's credit, they have taken both speed and standards compliancy to heart with their upcoming Internet Explorer 9. The SunSpider result graph in Figure 4, taken from their recent tests, indicate how their various releases of IE9 Platform Preview compare

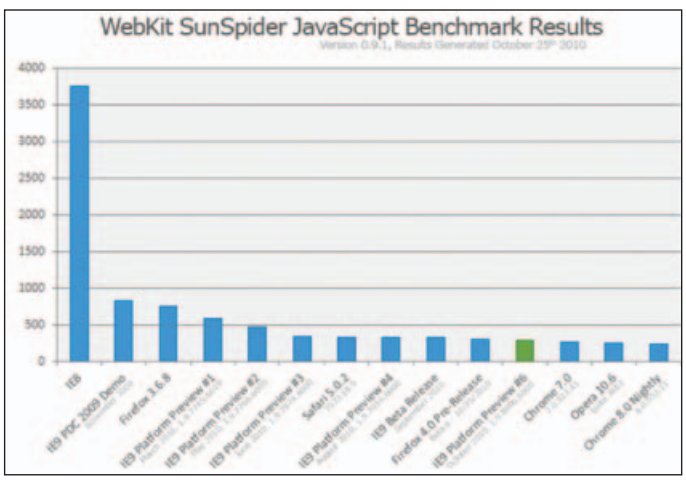


Figure 4. Benchmark results comparing various browser's JavaScript performance (lower is better)

to the older IE8 as well as the current crop of competitors. The latest pre-release of IE9 is 1,282% faster than IE8 – what a difference!

While those improvements are remarkable and give us hope for the future, using an unfinished Platform Preview release in a production environment is not at all feasible. If your customers and end users are still using IE6, 7, or 8, wouldn't it be great if there were a simple, free way to give them a massive boost in speed and throw in support for progressive technologies like HTML5 and CSS3? It turns out that is not only possible, but only takes a minute to download and install via Google's Chrome Frame.

Google's Chrome Frame for Internet Explorer

To ameliorate the performance and compliance problems with the older versions of Internet Explorer (IE), Google released the Chrome Frame. Their Web site code.google.com/chrome/chromeframe/ aptly sums it up with the following text:

Google Chrome Frame is an open source plug-in that seamlessly brings Google Chrome's open web technologies and speedy JavaScript engine to Internet Explorer. With Google Chrome Frame, you can:

- Start using open web technologies – like the HTML5 canvas tag – right away, even technologies that aren't yet supported in IE 6, 7, or 8.
- Take advantage of JavaScript performance improvements to make your apps faster and more responsive.

BASIS engineers added special code to the BUI system to automatically take advantage of Chrome Frame in IE if it is installed. This means that once Chrome Frame has been installed as a plug-in for IE, then running a BUI app in IE is just like running it in Google Chrome – it's fast and renders more accurately – all without the end user or BBx programmer having to do anything extra.

BASIS also modified their online documentation to use the Chrome plug-in, when available, so that the documentation will render as quickly as possible. If you attempt to run a BUI application in a version of IE that does not yet have the Chrome Frame installed, the BUI system will bring up the screen in Figure 5 to inform the user and facilitate the installation.

Summary

Web browsers are now a vital part of virtually all desktop and handheld computers. With BUI, BASIS extends its promise of BBj's platform portability by running on an unprecedented number of platforms and devices. As browsers vary by manufacturer, platform, and device, the differences may become evident and have an impact on the performance and look and feel of your application running in BUI.

While most of these differences are minor, speed is definitely worth looking into, as running your application on a slow browser instead of a fast browser is analogous to running your application on a slow computer instead of a fast one.

Now that the browser serves as the operating environment, choosing the right browser in which to run your application may mean the difference between a snappy, responsive application and a slow, lethargic one. As evidence of its commitment to the wide adoption of BUI, BASIS continues to test BUI applications on multiple browsers, and has made efforts to include support for Google's Chrome Frame plug-in in order to work around slower browsers. ■

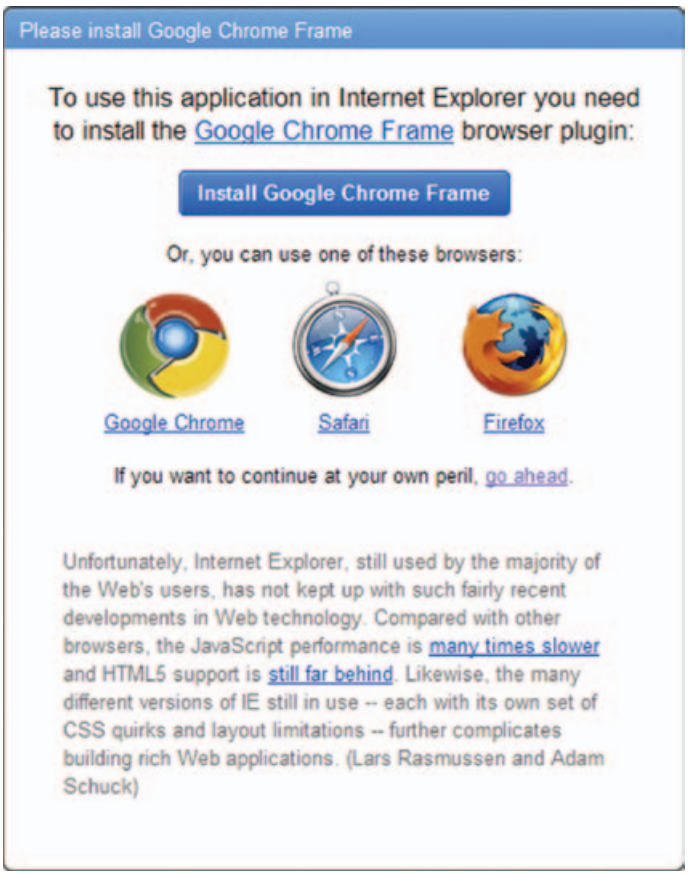


Figure 5. The screen prompting the user to install the Google Chrome Frame for IE

Google Chrome Frame
www.google.com/chromeframe

By **Jason Foutz**
Software Programmer

How BBj creates PDF

PDF (Portable Document Format) is the industry standard for distributing documents to users on virtually any operating system. BASIS now embraces PDF technology for creating documents and reports as all three BBj® print API's support PDF creation.

- SYSPRINT PDF leverages those reports that have been around for years
- BBjForm PDF provides a higher level API for creating documents
- BBJasperReport Utility allows quick binding of an SQL query to a report

SYSPRINT

The most exciting and probably the simplest PDF feature to implement is SYSPRINT support for PDFs. Many applications have reports specially customized for your users. Years of work have gone into creating these reports and tuning them to precisely match user requirements. Wouldn't it be great to output these same reports without having to change any code? Now, with PDF printing, you can create and save documents without having to change existing program logic. Using new PDF functionality only requires the addition of modes to your open statement or to the printer alias. For example,

```
open(rpt,mode="pdf,file=/home/report.pdf")"LP"
```

or in the config.bbx file

```
ALIAS LP SYSPRINT "printer" PDF,FILE="/home/report.pdf"
```

You may specify the mode either in the open statement in your program or in the alias for the printer device in the config.bbx file to avoid needing to change any existing application code.

BBjForm

BBjForm provides a higher level API for creating documents and reports. Whereas SYSPRINT depends on specifying the location of text character by character, BBjForm has a text layout engine to smoothly wrap words from one line to the next without the need for positioning mnemonics. BBjForm's higher level API frees the developer from much of the text layout details that return little benefit for the amount of time invested. Features like fitted fields, custom paragraph alignment, and drawing functions facilitate the creation of professional looking forms. The ability to create a BBjForm as a PDF is the icing on the cake, allowing developers to save a report to the portable file format just as easily as printing it out.

BBJasperReport

Finally, BBj provides a BBJasperReport utility that is based upon the world's most popular open source reporting engine, JasperReports. JasperReports streamlines the process of creating documents that use data sources for populating fields and rows of data. Use the JasperReports family of tools, e.g. iReports, to design your document graphically and then connect that report to a database. Once your report is ready, you can easily create a PDF with the API call `BBJasperReport::exportToPDF`.

Summary

Committed to the industry standard PDF file format, BASIS greatly simplified creating documents and reports for your users. Proceed with confidence that users can view your reports exactly as you create them. Whether you turn SYSPRINT reports into PDFs, create PDFs with BBjForm, or use BBJasper to create dynamic reports graphically and backed by SQL queries and/or SPROCs, all of these options support a variety of users … Perfectly Displaying Formats! ■

For more information on

- JasperReports, visit jasperforge.org/projects/jasperreports
- Report writing, read the Advantage article Recipes for Successful Report Writing links.basis.com/reportwriting

By **Chris Hawkins**
Software Programmer

Plumbing the Barista Framework Into BBj Forms

One of the powerful features in the Barista® Application Framework menu system is the capability to run programs other than Barista forms, making it possible to run your hand-crafted or AppBuilder/FormBuilder-crafted custom BBj® forms from within the Barista MDI. This article reveals how to incorporate Barista menus and toolbuttons into your custom forms to extend functionality and provide a consistent Barista look and feel. Adding this functionality to your own forms is just one of the ways you can offer a hybrid solution in Barista, delivering Barista form and function now in your custom forms without waiting to re-design your forms in Barista.

Background

Barista uses the group namespace for communication between the MDI and forms running within the MDI. A setCallbackForVariable() event registered on a namespace variable corresponding to a given form/task allows Barista to intercept and process MDI-level menu and toolbutton selections. At the form level, when you opt to place menu and/or toolbuttons on the form itself using bam_controls.bbj, you register callbacks for those particular items as well. This Barista infrastructure gives you the ability to recognize/intercept menu/toolbutton events in the custom BBj forms and process them accordingly.

In addition to using the namespace, plumbing Barista functionality into the custom BBj forms requires the following Barista public programs (publics):

- bac_mdi_ctls; builds Barista system variables containing control ID's and menu/toolbutton indicators for the various MDI menu and toolbuttons
- bam_enable;
  - initially sets which menu/toolbuttons should appear
  - toggles enable/disable status of selected menu items/buttons as form runs
- bam_attr_init; gets Barista attribute arrays used in other calls
- bam_controls; places menu and toolbuttons on form itself (rather than just MDI)
- bac_winsize; gets/saves form location and size from Barista settings file

These publics are written in BBj and may be called by programs written using other tools available from BASIS. AppBuilder/FormBuilder projects are easily integrated into Barista using this process.

Example - Customer Form

Our example is based on a simple customer form written in BBj but outside of the Barista Application Framework. The form allows basic add, change, and delete operations on a customer table. While you can launch the form easily via the Barista menu, the functionality is limited and the look and feel is inconsistent with other Barista forms. We'll see how to plumb some Barista code into the form so we can obtain a Barista look and feel while intercepting and processing menu and toolbutton events from both the form and the MDI.

The customer "form" is really two files: an ASCII resource file (.arc) that describes the physical characteristics of the form, and a BBj program (.bbj) that reads/displays the .arc file, controls file I/O, data input, event handling, etc. Figure 1 illustrates how the form looks before we incorporate the Barista menu/toolbuttons:

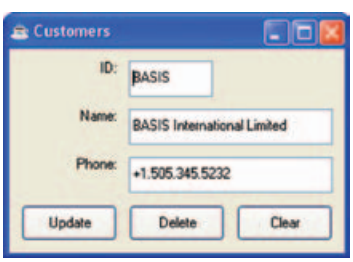


Figure 1. Customer Form without Barista menu/toolbars

The program registers callbacks when we close the form, edit or lose focus in the customer ID field, or push any of the three buttons. However, there is no record navigation (first, last, previous, next) and you must know a customer ID in order to call up any given record.

To provide Barista look, feel, and functionality to our form, we'll make modifications to both the .arc and .bbj files. In the .arc, we'll delete the buttons we no longer need and change the window control ID from 101 to 1000. Barista uses (and expects) certain controls to have IDs in pre-defined ranges. Developers rarely need to concern themselves with control IDs when designing forms within the Barista Application Framework. However, if we are integrating Barista with an "outside" form, we must examine and alter the various control IDs in use to avoid conflicts.

The original .bbj program contained callbacks for the Delete, Update, and Clear buttons that routed to corresponding subroutines. We'll keep those routines in the modified program, but now they will be executed as a result of toolbutton or menu events. Likewise, we'll keep the routines used to read and display records, but where the original program only read/displayed records based on user input, the new program can execute that logic as we use Barista navigation buttons or corresponding menu selections.

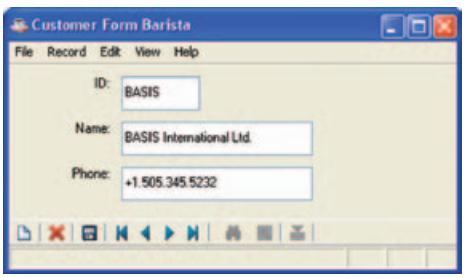


Figure 2. Customer form revised to include Barista menu/toolbuttons

In the old form, as we began new data entry, we had specific code to disable the Update and Delete buttons. In our new form (see Figure 2), we provide the same functionality by just passing the control ID for the desired menu item or toolbutton into the Barista public that enables/disables controls.

The heart of the Barista processing is shown in Figure 3, in the code that is executed as we analyze which menu item/toolbutton has been selected and then route control to the appropriate routine.

Summary

By incorporating the Barista menu and toolbutton functionality into the customer form, we can rather quickly provide a hybrid solution, running our new form seamlessly with other Barista forms without an extensive rewrite of our existing custom code! ■

```
route_func:rem --- Get Function from form-based toolbar/menubar
rem --- callback to this routine was registered in the
rem     call to bam_controls.bbj.
rem --- this routine runs when clicking a menu item or
rem     toolbutton on the form.

   evtQueue!=sysGUI!.getLastEvent()
   tempCtl!=evtQueue!.getControl()
   active_func$=tempCtl!.getUserData()
   func_source$="FRM"

   goto route_active_func

get_active_func:rem --- Get Function from MDI/Namespace
rem --- this routine runs when clicking on a menu item or
rem     toolbutton on the MDI.

   active_func$=""
   active_func$=sysGUI!.getLastEvent().getNewValue(err=*return)
   if active_func$="" return
   grpSpace!.setValue("+"+task_val$+".func","")
   func_source$="MDI"

route_active_func:rem --- Route Function to appropriate subroutine
rem --- after executing route_func or get_active_func, use
rem     this routine to analyze the function and route accordingly

   func_str$="EXT;NEW;SAV;DEL;FST;PRV;NXT;LST;"
   switch fnstr_pos(active_func$,func_str$,4)
      case fnstr_pos("EXT",func_str$,4)
         gosub exit_prog
      break
      case fnstr_pos("NEW",func_str$,4)
         gosub clear_frm
      break
      case fnstr_pos("SAV",func_str$,4)
         gosub update
      break
      case fnstr_pos("DEL",func_str$,4)
         gosub remove_rec
      break
      case fnstr_pos("FST",func_str$,4)
         id$=keyf(customer,err=*break)
         gosub fetch_nav
      break
      case fnstr_pos("PRV",func_str$,4)
         read record (customer,key=customer.id$,dir=0,err=*break)
         id$=keyp(customer,err=*break)
         gosub fetch_nav
      break
      case fnstr_pos("NXT",func_str$,4)
         id$=key(customer,err=*break)
         gosub fetch_nav
      break
      case fnstr_pos("LST",func_str$,4)
         id$=keyl(customer,err=*next)
         gosub fetch_nav
      break
      case default
      break
   swend
   return
```

Figure 3. Sample of route_func: and get_active_func: routines catching menu/toolbutton events from form or MDI, respectively, and route_active_func: routine handling the actual routing

For more on this topic, including code samples and an additional Barista example, check out "Barista Plumbing Exposed!" links.basis.com/nfrim

By **Laurence Guiney**
Senior Account Manager

BUI to The Rescue

Recently, two unrelated concerns surfaced that BASIS was able to address with one very exciting solution.

Can BASIS provide a real-world working example of the new Browser User Interface rather than the typical demo application with limited functionality? Can BASIS make their really useful b-commerce® application more accessible to me by not requiring that I install a JRE? Read on to see how BUI (Browser User Interface) came to the rescue!

An example of a BUI conversion

The code examples that BASIS provides for most new features adequately illustrate their application, however BUI is a major game-changer with a very widespread impact. BASIS invested a great deal of time and engineering resource developing this interface. The BUI example really needed something bigger.

Most BASIS customers are already familiar with b-commerce. BASIS chose this application to demonstrate just how easy it is to turn a GUI application into a BUI app. This was just the real-life BUI example BASIS customers were asking for and would hopefully lead them on their way to exploiting the power of BUI.

Making b-comm more accessible

In 2000, BASIS used Visual PRO/5® to develop b-commerce as their internal order placement system. Later, BASIS converted it to BBj®, making it an external application for their partners to access via the Web using BBj's thin-client technology. Today, customers use this application to place their BASIS orders.

While b-commerce is a powerful application, many have asked how to access it without the installation of a JRE on the client computer, which is sometimes helpful or necessary at an end-user location. The solution was clear. BASIS' wafer-thin client – the Browser User Interface, simply known as BUI – would allow such access without the need for a JRE.

The Rescue

Moving this GUI application to BUI allows to run the application on any JavaScript-enabled browser. This was a working example of taking GUI to BUI while eliminating the need for the installation of a JRE.

It took little effort to get b-commerce up and running as a BUI application. The major task was reviewing the controls used and verifying that they were available in BUI. Since all the controls used in b-commerce are available in BUI, few changes were necessary.

The next step in the process was testing and optimizing. BASIS employees who run b-commerce as part of their daily responsibilities tested the various functions. Once the testing and optimizing was complete, BUI b-commerce was ready for use by all BASIS customers.

The Launch

Earlier this year, BASIS delivered BUI b-commerce to our customer base. The initial feedback was very encouraging.

Summary

Just as BASIS has encouraged their customers to invigorate their old apps, so has BASIS taken their own advice. BASIS migrated a GUI application developed in Visual PRO/5 to a BUI application. Customers wanted to see a real-world working example of BUI so BASIS gave them one. At the same time, this BUI migration also made the heavily used online order and information application accessible on almost any convenient browser, without needing a JRE, to all customers around the world.

BUI provided BASIS with the solution; perhaps BUI may provide the rescue line you need. ■

To try BUI b-commerce, go to www.poweredbybbj.com and enter your customer number and password. If you are not set up to use b-commerce, contact your BASIS sales representative for assistance.

By **Jason Foutz**
Software Programmer

BBj's Web Integration is Better Than Ever

Like a fine wine, BBj gets better with age as does the Jetty Web server integration! BBj now supports SSL connections for all types of Web connections, be it Browser User Interface, Web Start, or simple file serving. More configuration options are now available in Enterprise Manager (EM). BASIS also added many user interface improvements for configuring Web Start and Web Services.

Secure Sockets Layer - Subtler

When delivering an application over the Web, SSL encrypts the data sent between Jetty and the Web browser, keeping your communication secure. By default, BBj installs with a sample certificate that encrypts communication with the client.

To use your own certificate with SSL, first obtain a certificate from a provider such as GoDaddy or VeriSign. Then configure Jetty to use this new certificate in two steps:

1. Create a new keystore using the following command:
   ```
   keytool -keystore keystore
   -import -alias jetty -file
   jetty.crt -trustcacerts
   ```
2. Use EM to configure BBj to use the newly created keystore.

After installing a proper certificate, users will no longer be warned about an untrusted or invalid certificate. Subtle and more palatable.

EM and BUI - Bigger Bouquet

Another new EM feature is the ability to view and edit Browser User Interface (BUI) application configurations, which provides a great degree of flexibility. You may configure a BUI application with an install program written in BBj or now use EM to select your config file and program as well as other options. EM then creates the application configuration. In addition, EM provides a centralized listing of all the installed BUI applications from which you can launch the applications or view their configuration.

Demos - Deeper

The internal demos are deepening with age, automatically updating hostname and port number as needed. For example, when changing the hostname in EM, that information immediately reflects in the demo Web Start application. Configuration settings for the Web Start demos are now separate, making it easier to enable or disable specific applications.

Interface - Fuller Body

The Web Services user interface is also improved, making it easier to configure applications to be accessible as a Web Service, even business logic in PRO/5® code can be offered up as a Web Service. The addition of tooltips and labels simplify the creation of new Web Services. When examining a built-in application like the BASIS Update Service (BUS), the improved user interface makes it easier to browse the application configuration or to view the available operations.

The Final Report

BASIS is committed to bringing your application to the Web by offering the premier interface of choice, BUI, now configurable via EM. The EM also makes it easy to protect your users' data using SSL protocol. Jetty enhances BBj, making your applications easier to configure, more secure, and more robust. BBj is clearer, the taste sweeter, its aroma more fragrant… a more well-balanced expression of its origin. ■

For more information, read

- Jetty Offers Legacy Programs via Web Services
  links.basis.com/jettywebservices
- Jetty Web Server for BBj, Java Style
  links.basis.com/jettywebserver

A Quarter Century of Progression .

BASIS 25th Anniversary — the next generation

BASIS Incorporated founded — January 1985

BASIS launches Business Basic Extended — October 1985

BASIS moves to electronic BBS — December 1986

BBˣProgression/2® — May 1987

New name and logo — January 1988

BBˣ Level 1 — October 1985

BITS Conference — November 1992

Coyote College Training — March 1993

BITS Conference — April 1993

BBXchange Conference — September 1993

TAOS/Views™ — March 1994

BITS Conference — November 1994

PRO/5® 1.0
Visual PRO/5® 1.0
PRO/5 Data Server® 1.0 — October 1995

BASIS ODBC Driver 2.0 — September 1998

PRO/5 2.0
Visual PRO/5 2.0
PRO/5 Data Server 2.0
ResBuilder® 1.0
GUIBuilder® 1.0 — June 1998

TechCon — September 1999

PRO/5 3.0
PRO/5 Data Server 3.0
BASIS ODBC Driver 3.0 — October 1999

BBj IDE, BBj 2.0 — April 2002

TechCon — May 2001

PRO/5 4.0, Visual PRO/5 4.0 — July 2002

BASIS products ship on CD — March 2003

Audev Germany partnership — April 2007

PRO/5 6.0, Visual PRO/5 6.0 — August 2006

BBj 6.0 — June 2006

TechCon — May 2006

20th anniversary BASIS products — October 2005

BBj 5.0 — May 2005

AddCon — November 2008

AddonSoftware® 8.0 — August 2008

PRO/5 8.0, Visual PRO/5 8.0 — June 2008

BBj 9.0, Barista 9.0 — March 2009

AddonSoftware 9.0 — July 2009

TechCon/AddCon — September 2009

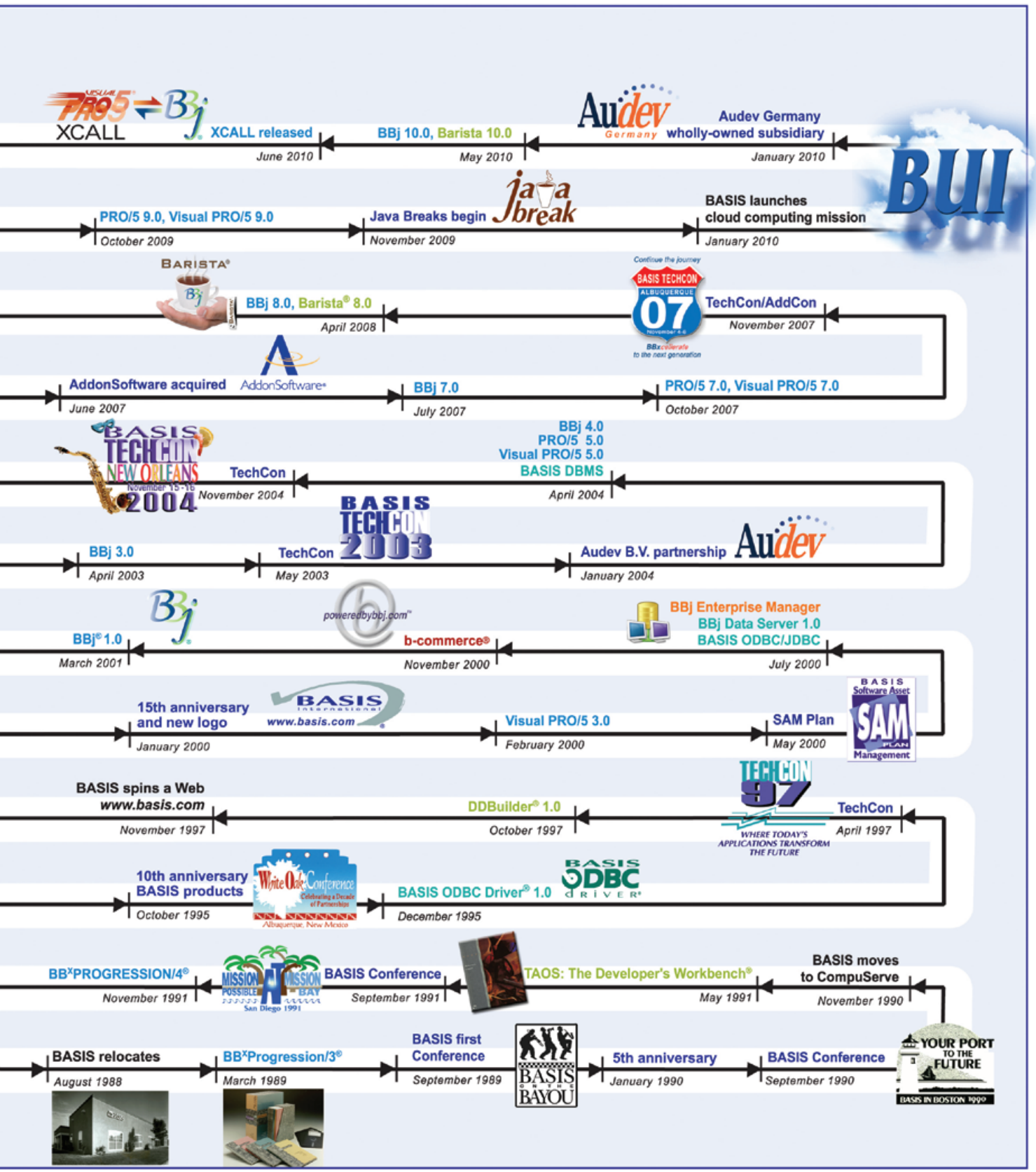
TechCon/AddCon — May/June 2011

5th BASIS products

BASIS/EMS Diamond Work Group Conference — October 2010

PRO/5 10.0
Visual PRO/5 10.0
AddonSoftware 10.0 — July 2010

...the BBx® Generations



By **Brian Hipple**
Quality Assurance Supervisor

Introducing the BASIS Custom Installer

Relying on a third party software product is generally very helpful and efficient, at least until the time when the product is no longer supported. We experienced this dilemma several months ago when we learned that our InstallShield installation product has reached its 'end of life.'

We could have ported our InstallShield project to a new product, but since we had already ported to two different installers that had both quickly became unsupported, we decided to take the opportunity to create a homegrown installer using our own in-house Java expertise. The result – an installer that precisely meets our needs and growing wish list, and has no risk of being discontinued.

Meet the BASIS Custom Installer, fondly referred to as the BCI, now available in the BBj 10.0 release.

What's New

The BCI retains all the bells and whistles of the previous installer. It was important to still allow users to choose the language, select an installation directory, identify which Java version to use, and to customize the product components to install.

Like its predecessor, the new BCI also runs with a Graphical User Interface (GUI) or in a Character User Interface (CUI) for those server machines that lack GUI support. This rewrite opportunity gave us the chance to make some important changes, improve performance, and add new features, all in a smaller footprint.

Start the Installation

The first change was to simplify the "start the installation" process. We combined the setup.exe for Windows, install script for UNIX, and command for Macs into one executable jar for all platforms. On most systems where jars are correctly associated with Java, simply double clicking the jar or selecting "Run" when downloading from the BASIS Web site starts the installation. On systems that do not have this association setup, just run the installable jar as follows:

```
java -jar <BASISInstallable.jar>
```

For example,

```
java -jar BBjBaristaAddon000_10-04-2010_0859.jar
java -jar BBjBaristaAddonIDE1002_09-25-2010_1546.jar
```

Silent Installation

Another enhancement in the BCI is in the silent installation process. While it still supports recording and playing back from a response file, it does so using a slightly changed format. The response file is no longer in proprietary InstallShield format, but a standard Java properties file as shown in Figure 1.

If you are currently using a silent install, you will need to re-record your response file into the new format. The process is the same and pain-free, so no need to fret. Just record the response file, edit the specific settings in the file using the editor in the BASIS IDE to edit key-value pair property files, then playback the response file. To facilitate the recording and playback of a response file, the first installation screen has a check box for recording and playing back for a specified response file. See Figure 2.

Specify the response settings on the command line as follows:

```
java -jar <BASISInstallable.jar> [-rpl] <InstallResponse.properties>
-r record
-p playback
-l lock response file settings, so that a user can not change them
```

For example,

```
java -jar BBjBaristaAddonIDE1000.jar
java -jar BBjBaristaAddonIDE1000.jar -r /tmp/InstallResponse.properties
java -jar BBjBaristaAddonIDE1000.jar -p /tmp/InstallResponse.properties
java -jar BBjBaristaAddonIDE1000.jar -lp /tmp/InstallResponse.properties
```

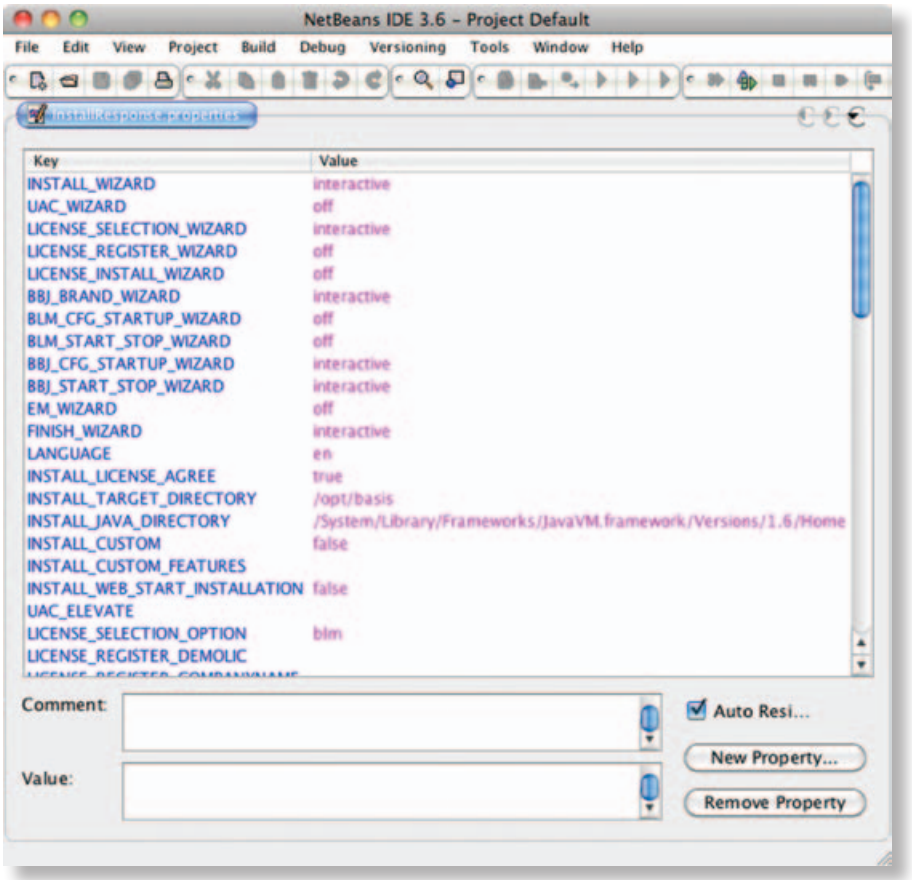


Figure 1. Example of a response file

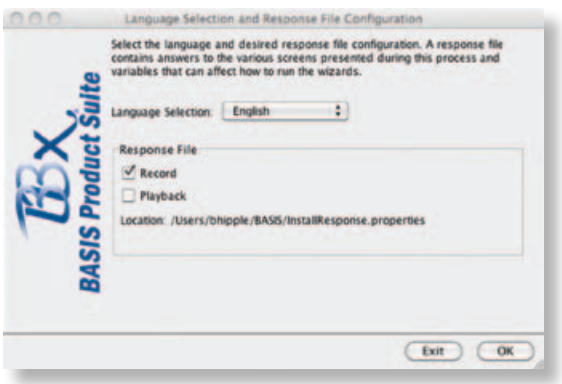


Figure 2. The installer's initial screen displaying response file options

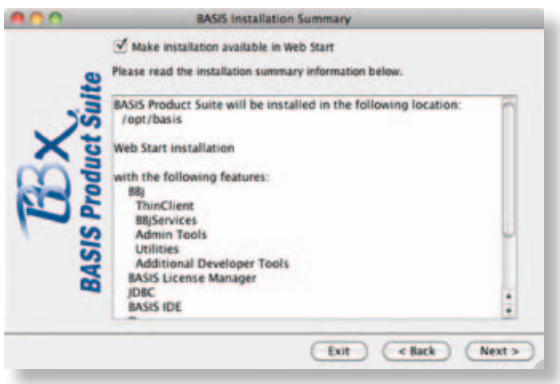


Figure 3. The summary screen displaying the Web Start Installation option

Size and Performance

We also made huge strides in improving both footprint size and performance. The installer is now almost a third smaller, depending on the selected components. This is especially advantageous when downloading the installation package via the Internet from the BASIS Web site. Also, the installation time can be up to five times faster than the previous installation, depending on the amount of memory and the I/O speed of the installation machine.

Web Start

One of the exciting new features in the BCI is the ability to serve the installation via Web start. This new option (see Figure 3) provides the ability to keep the server and clients updated to the same version.

Clients can upgrade themselves by simply clicking on a URL inside an e-mail or on a Web page. For example, if you just upgraded a server, you can easily update all client BBj ODBC drivers by sending an e-mail to your users, instructing them to click the link to start the installation (perhaps silently?). Imagine the potential time savings … and time is money!

How exactly does this work? The installer copies the installable jar into the target installation directory, where the Jetty Web Server portion of BBj Services serves up the Web Start version of the installer. You can specify an optional response file and customize the link. To configure the Web Start installation, select the JNLP configuration and install node in the Enterprise Manager as shown in Figure 4.

Summary

BASIS designed the BCI to be a very generic installer using an instructional XML file that tells the installer what files to install by component, including overwrite instructions, defaults, and much much more. This instruction file is easily manipulated, therefore, it's possible to open up the installer to install any application, even yours. Take a few minutes to check out the utilitarian functionality of the new BCI and see how it can help you and your company work smarter, not harder. ■

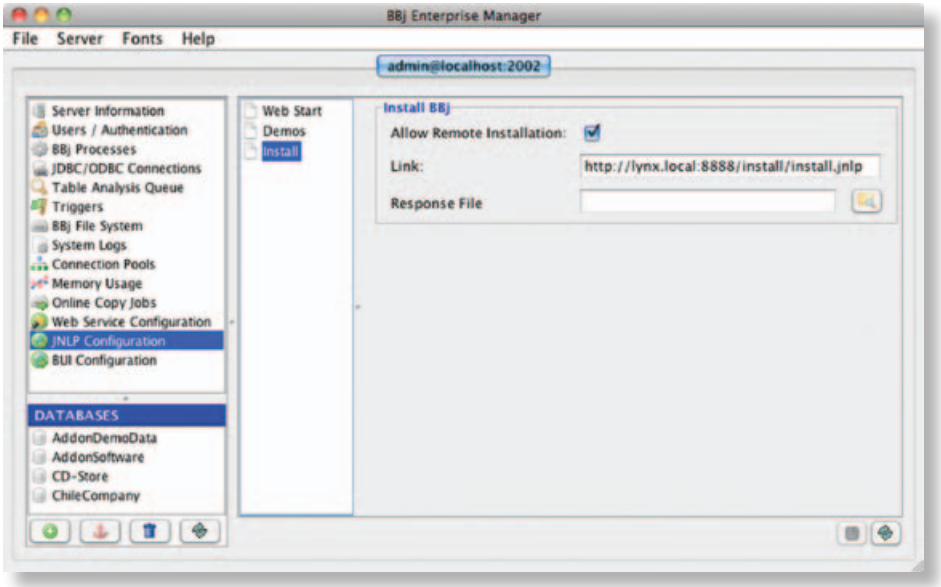


Figure 4. Enterprise Manager's Web Start Installation configuration

"What are these quaint little dots about?"

In this "Advantage", you come across so-called QR-Code.

"QR" stands for „Quick Response". It is a barcode which you can read with your camera-enabled mobile phone by just pointing your camera eye at it, saving you the labor of typing the URL.

If your phone is equipped with QR code decoding software, it will fire up your browser and go straight to that URL which provides deeper information on the subject at hand.

It also works for contact data, e-mail-addresses and much more.

Let us know what you think about this new BASIS service!

More information on QR code and free QR decoding software for your smartphone can be found here:
http://www.qrstuff.com/qr_phone_software.html

By **Jeff Ash**
Software Engineer

Automatic Database Analysis

Dramatically boost the performance of your queries

Automatic database analysis is a powerful feature that removes the necessity of a BBj® Services administrator to explicitly run a database analysis operation on their database. Database (or table) analysis is a feature where the BBj SQL engine analyzes the tables in the database to generate information used for determining the best possible way to optimize SQL queries run against that database.

Without this information, the SQL engine must make a generic "guess" as to how to optimize a query and may or may not choose the best option. SQL query performance is highly dependent upon the availability of this information.

How It Works

The BBj database engine examines tables each time they are accessed to determine if an analysis is warranted. The following criteria is currently used to determine if a table should be analyzed or reanalyzed:

- The table has not yet been analyzed
- Since the last time the table was analyzed
  - The record count has increased by more than 50%.
  - One or more indices were created or dropped from the table (also includes keys added or removed from the file using non-database operations or tools such as SQL or the Enterprise Manager)

If any of these criteria are met, a table analysis job is entered into the analysis queue (see Figure 1) where it starts as soon as there is an available time slot for it to run. The analysis engine intentionally limits the number of simultaneous analysis processes so that it does not interfere with the running of applications or queries on that server. To view the progress of analysis jobs and the order of queued tables, simply click on the "Table Analysis Queue" item in the navigator on the left side of the Enterprise Manager application window.

How it Defaults

An important consideration is whether to leave automatic table analysis enabled for a database or if it should be disabled and analyzed manually. In most cases, automatic analysis should simply be enabled, which is the default setting on setup of a new database in the Enterprise Manager. However, if your application has the occasion to remove data files or tables that are part of the database, it may be necessary to disable automatic table analysis since an analysis operation will prevent a file currently being analyzed from being removed.

In future releases of BBj after 10.02, developers can use the Admin API to disable/enable table analysis to avoid the need to completely turn off auto analysis.

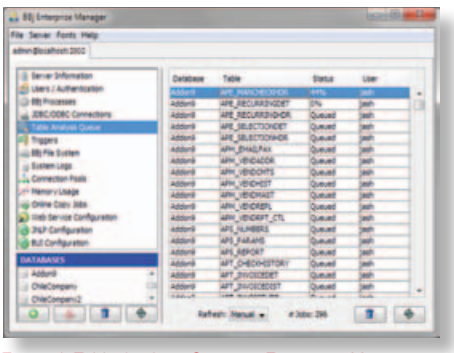


Figure 1. Table Analysis Queue in Enterprise Manager

This feature allows the application to disable analysis if it needs to manipulate the files in an exclusive way, whereby database analysis might interfere.

Summary

Database analysis is vital to achieve optimal performance of SQL statements in BBj. Without the information derived during analysis, the SQL engine has no way of knowing the best index to use when matching a WHERE clause and so it must simply guess. With the addition of automatic database analysis, there is one less application setup action required to get a database configured and keep it configured to run as fast as possible. Now, BBj automatically analyzes tables as the need arises, all without any interaction from the end user or the administrator. ■

By **Paul Yeomans**
Vertical Market Account Manager

AddonSoftware's Barista Building Blocks Strategy

AddonSoftware provides out-of-the-box functionality without the constraints of the box.

One might ask, "Why would a software language and tool developer take on an ERP solution"? The answer lies in understanding the solution's origins and our strategic vision.

AddonSoftware® is a proven and reliable ERP solution, originally developed in the early 1980's by MicroAccounting Systems Inc. (Beaverton, OR) and more recently enhanced using ARDE (Addon-Rapid-Development-Environment), the predecessor of what became our powerful Barista® Application Framework development tool.

Meanwhile, we witnessed the successes of our partners using our technology to develop their own verticals and core accounting solutions.

AddonSoftware provided us with an opportunity to take the lead when we acquired the solution in 2007. Combining our strong technical skills with our own tool set – the BBj® "Java Made Easy" language – and the high-powered Barista application development tool, we continued to overhaul and modernize the original code resulting in AddonSoftware by Barista.

The Strategy

Imagine standard modular home construction which includes a kitchen, dining area, bathrooms and a few bedrooms. Now imagine the ability to add a sunroom, a wine cellar, a pool in the yard, or a home theater to the standard home. In addition to choosing new rooms and features, you have the opportunity to update the look of any of the basic rooms to suit your unique needs and taste.

This is the concept behind our building blocks strategy. AddonSoftware provides out-of-the-box functionality without the constraints of the box. BASIS, through AddonSoftware, provides the core building blocks common to modern ERP solutions, around which you can easily build your own customized solution.

Though we provide a virtual shrink-wrapped ERP solution, we also recognize that a one-size-fits-all business solution fails to address the unique practices and requirements of a particular business. Meeting these unique practices and requirements provides value and utility to end users and necessitates customization. However, your prospects

may be averse to a customized business solution since they are typically very complex and costly. Furthermore, customization often leaves solutions "frozen in time" and unable to upgrade to take advantage of new features without extensive re-writing of the custom code (see Preserving Your Customizations on page 30).

AddonSoftware is the Answer

Neither overly complex nor costly, and able to preserve your customizations through the upgrade cycle, AddonSoftware is the answer. AddonSoftware resolves the dilemmas posed by most customized business solutions by combining "out-of-the-box" functionality with the Barista tool to facilitate your application development.

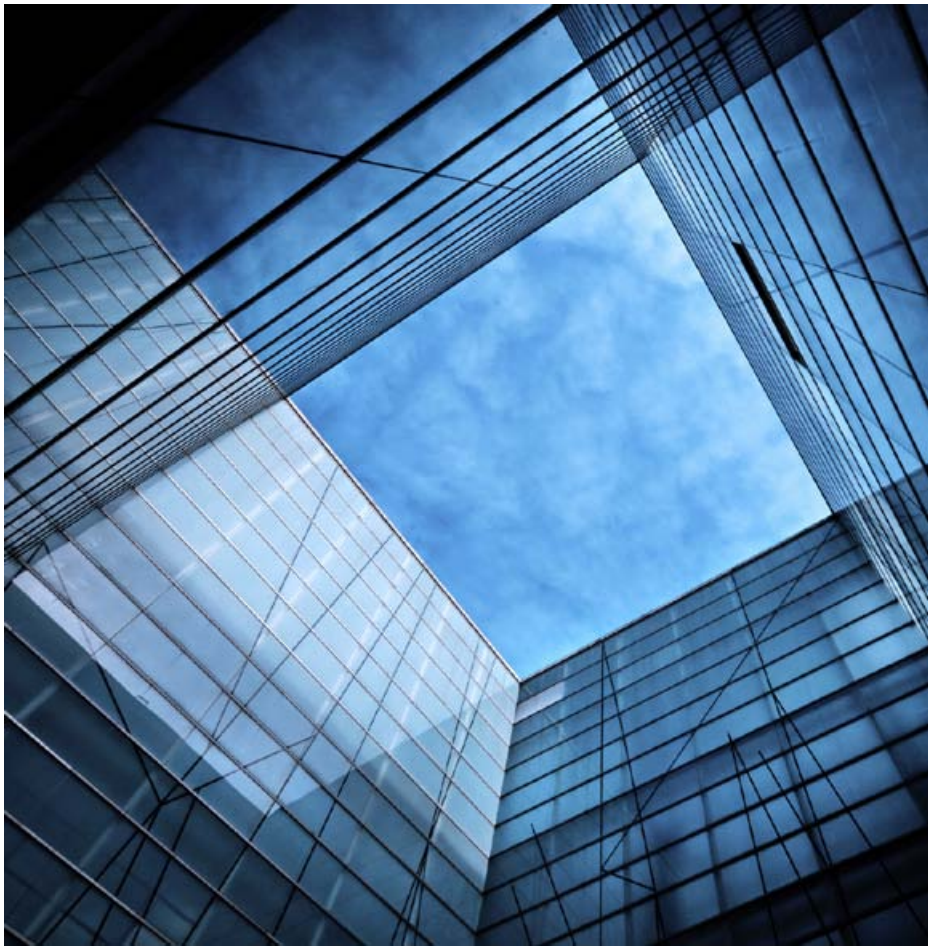
No hidden costs means that we include everything you need for one price; the BBj Enterprise Edition, the Barista development tool, and a long list of utility features that may replace the third party vendors in your current solution.

Combined with the efficiencies of the development tool and the ease of future maintenance tasks, AddonSoftware's core building blocks gives you the opportunity to lower the cost of your next project and the time to make more sales.

Simple Selling

Selling AddonSoftware is simple. It comes in three integrated building blocks or bundles – Accounting, Distribution, and Manufacturing. Each bundle contains related modules and may contain other bundles. Accounting is the base bundle that includes General Ledger, Accounts Receivable, and Accounts Payable.

The Distribution bundle starts with the Accounting bundle and adds Sales Order Processing, Purchase Order Processing, Inventory Control, and Sales Analysis modules. The third bundle, Manufacturing, starts with both the Accounting and Distribution bundles and adds Shop Floor Control, Bill of Materials, and Material Requirements Planning. Custom construction



allows you to choose your room and customize your design. AddonSoftware gives you that same freedom through your ERP solution.

Unique Opportunity

Membership in the AddonSoftware Partner Program provides a unique opportunity to become involved in the creation and further enhancement of the product.

We took a page from successful open-source models but added a commercial twist. Our community contributes to the development of the product and receives rewards with product credits for their efforts, leading to margins as high as 100% – better than open-source!

This unusual approach allows you to benefit from the expertise of others while jointly extending the range of the solution.

The Opportunity for You

With BASIS as the facilitator and deliverer of the building blocks, and you as the Master Craftsman, there are no limits to how far you can grow your business.

AddonSoftware partners enjoy generous margin opportunities and the technical and training support you would expect from a long-standing industry leader of software development tools. ■

Interested in an AddonSoftware partnership? Contact us today at info@addonsoftware.com.

By **Chris Hawkins**
Software Programmer

Preserving Your Customizations

"With Barista, you can customize applications in a way that preserves modifications through upgrades of the base product."

Like snowflakes, no two businesses are exactly the same and neither are their information processing requirements. How then, does one justify selecting a standardized off-the-shelf package to meet businesses' increasingly diverse IT requirements?

One argument is that custom software is often more expensive throughout its entire life-cycle than a packaged solution, but packaged solutions are almost always inadequate in some areas. Up-front savings in a packaged solution can quickly dissolve as end users or IT staff have to devise external processes to make up for the package's shortcomings.

On the other hand, customizing a packaged solution can leave users "coded into a corner," rendering the package ineligible for future upgrades and enhancements without losing the customizations.

If you are nodding your head in agreement and want to have the best of both worlds, read on. The Barista® Application Framework provides an answer to the dilemma! With Barista, you can customize applications in a way that preserves modifications through upgrades of the base product.

Barista keeps track of customizations by saving them in a special project file structure outside of the base product's install location and then re-incorporates them after a product upgrade.

This article goes step by step through the customization and re-installation process using AddonSoftware® as an example.

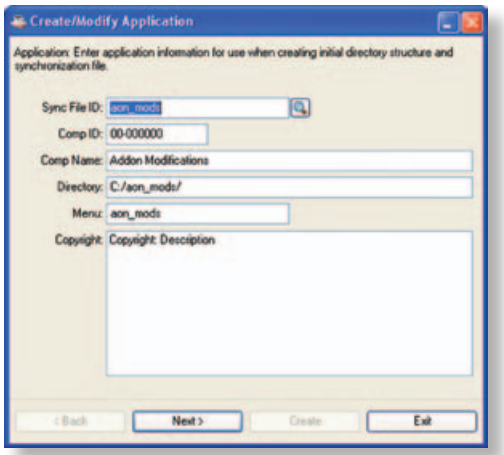


Figure 1. Create Application wizard collects information for the project

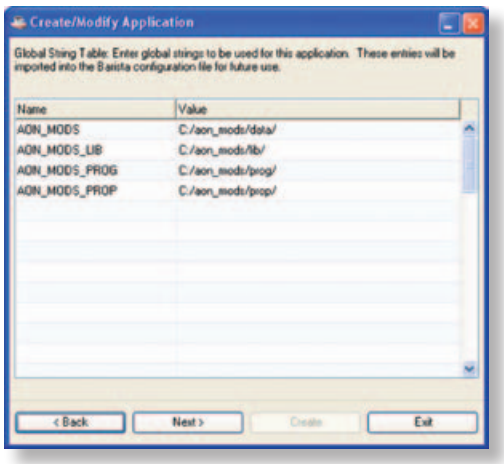


Figure 2. Barista provides default configuration settings

Step 1: Creating an Application

Before making modifications to a Barista application, you need to set up a directory structure in which Barista will save those modifications. Barista's Create Application wizard, as shown in Figure 1, collects information about your project such as the top level directory, your company ID, a description for the project/application, etc. and creates an application-area for the modifications.

Barista uses the information you provide to create default STBL values for the barista.cfg and BASIS config.ini files (Figure 2):

When the wizard has collected all necessary information, Barista creates a file structure for your project and prompts you to run the Auto-Synchronization process (Figure 3). This process forwards information provided in the wizard into your barista.cfg and BASIS config.ini files (Figures 4a and 4b), and you're ready to begin making customizations.

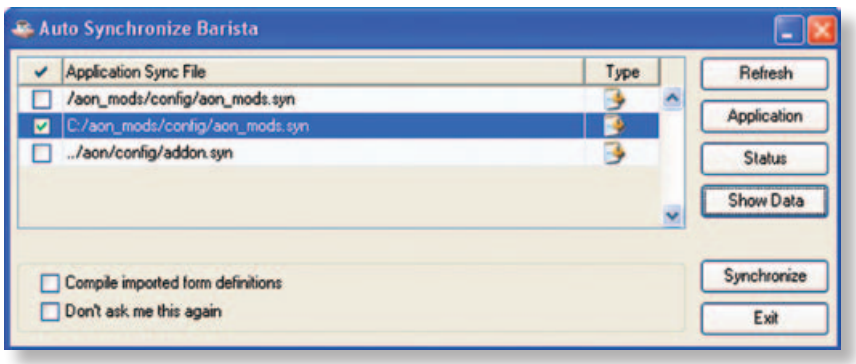


Figure 3. Auto-Synchronize new project into Barista

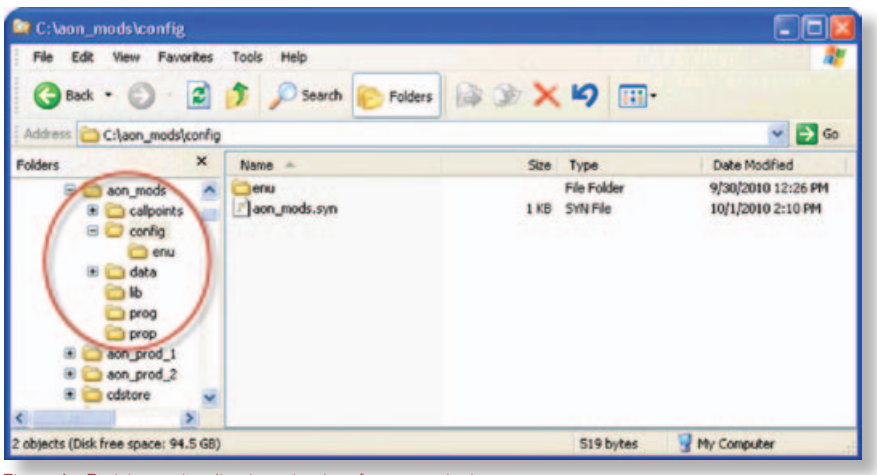


Figure 4a. Barista creates directory structure for new project

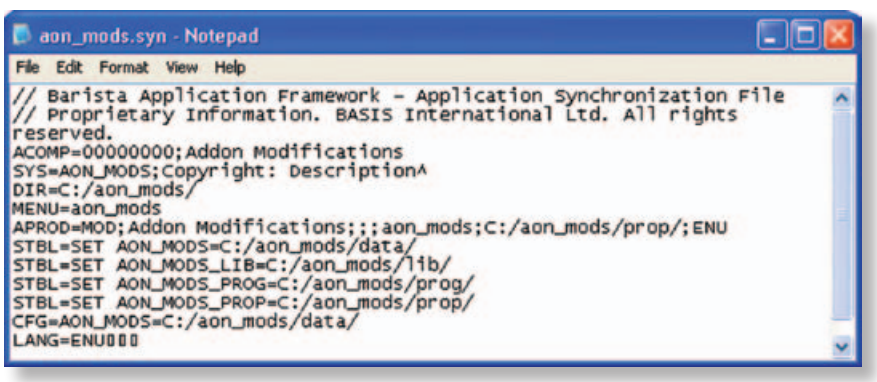


Figure 4b. Information supplied during wizard is captured in .syn file for new project

Step 2: Customize the User Interface

In order to customize forms or business-logic callpoint code, call up the Form Manager and right-click on your project name. The padlock icon that appears over the top of your project icon (see Figure 5) indicates that Barista is in "replication mode." Any form or callpoint changes you make to the base product will also be saved into your project area.

First, we'll create a brand new table and form directly in our project area to store new Vendor Category codes and descriptions (i.e., categorize what we purchase from this vendor: raw materials, inventory items, services, etc.). Then we'll alter the standard AddonSoftware Vendor Master form, adding a Vendor Category field that validates to our new table as shown in Figure 6. When we save and build the form, the Barista resource file (.xml) is saved in the standard product's `/data/bar` folder, as well as in your project's `/data/bar` folder.

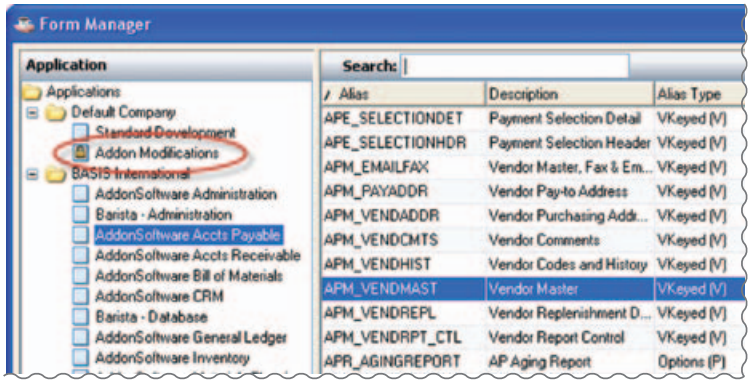


Figure 5. Development in replication mode

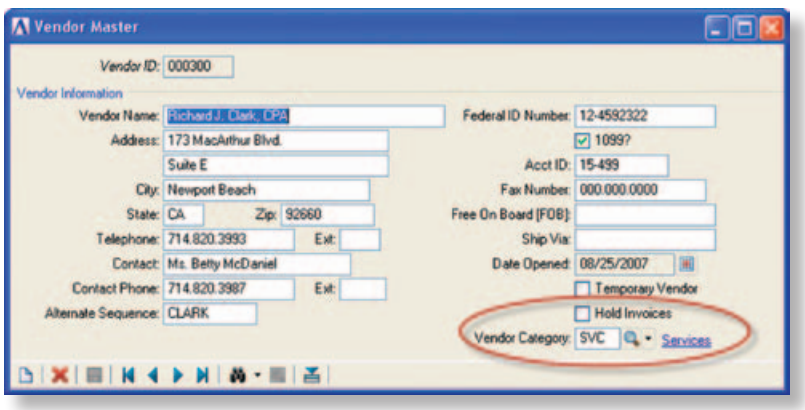


Figure 6. Revised Vendor Master form: moved Hold Invoices to the right and added new Vendor Category field

Step 3: Add Custom Business Logic

In order to make sure that we always have a value in the Vendor Category field, we can also add custom callpoint code. In our example, we set the Vendor Category to "UND" if no code is yet defined.

When in replication mode, you can see the callpoint code for the standard product, but can't modify it directly. Instead, you add code that executes before, after, or instead of the standard callpoints. Barista runs the Before callpoint code, then any code that's part of the standard product, then the After callpoint code (see Figure 7). To run your code only, that is, instead of the standard callpoint code, use the Before callpoint along with the method `callpoint!.setStatus("SKIP")`.

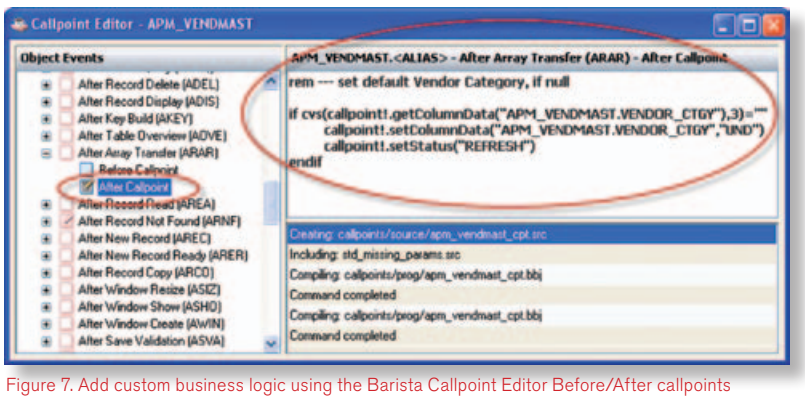


Figure 7. Add custom business logic using the Barista Callpoint Editor Before/After callpoints

Step 4: Create Custom Reports

In addition to modifying the user interface and business logic controlling the forms, you can also customize or write new "back-end" code such as reports, updates, or publics. Extending our AddonSoftware example, we'll create a modified version of the Vendor Name and Address listing that also shows our new Vendor Category. First, we need to alter the "Run Program" setting in the Option Entry form for the report so that Barista runs our customized report as shown in Figure 8.

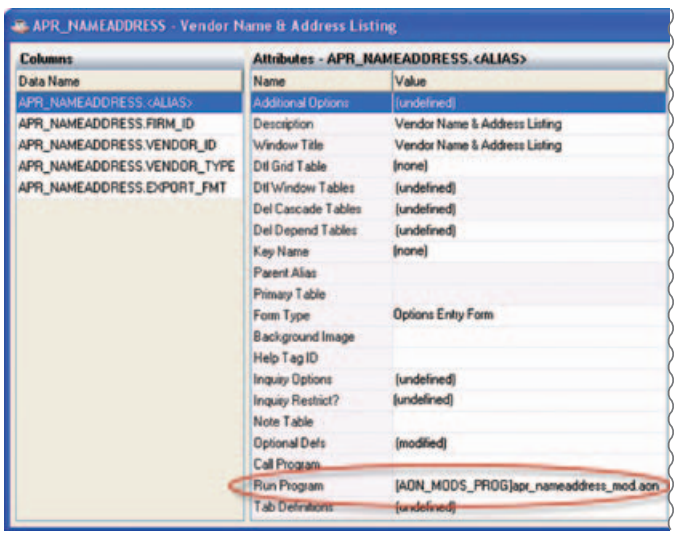


Figure 8. Changing the "Run Program" tells Barista to run our customized report

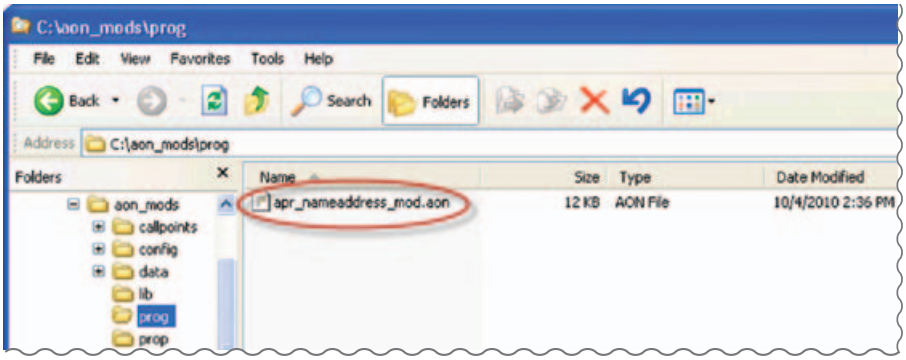


Figure 8a. Place back-end code in your project's "prog" directory

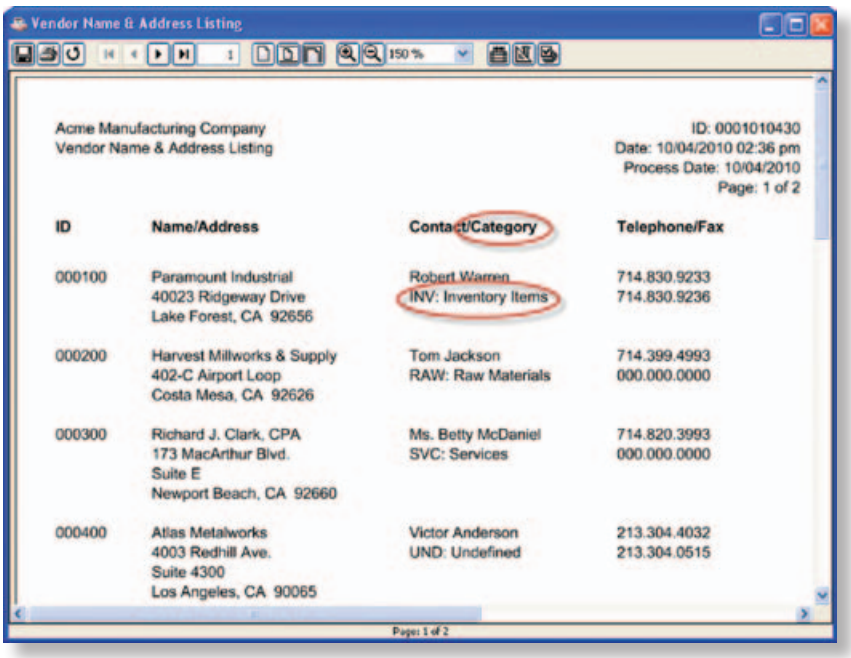


Figure 9. Customized Vendor Name and Address Listing

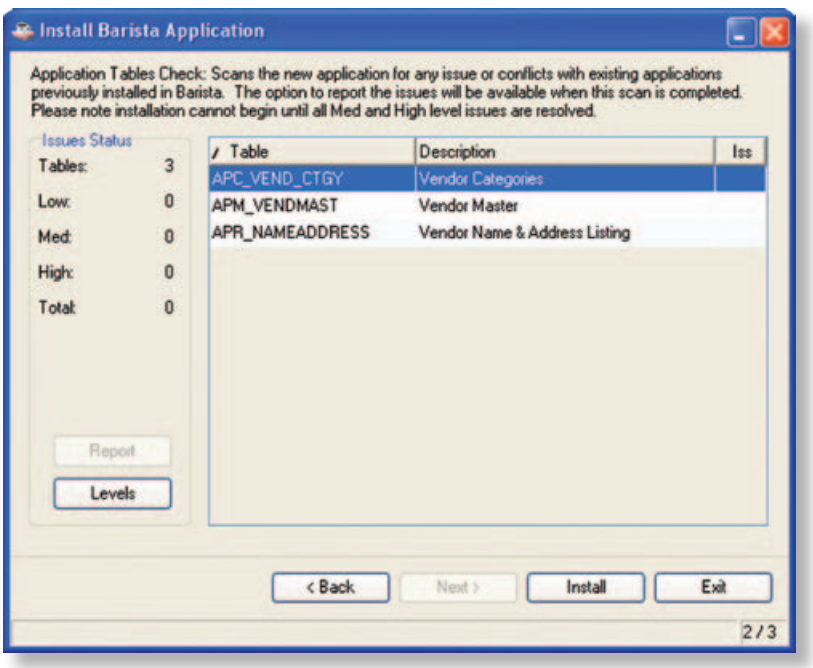


Figure 10. Barista Install Application Wizard

Remember, because we're in Replication mode, the Barista resource file for the Vendor Name and Address listing will be saved into our project area for safekeeping.

Next, we'll use a text editor such as the BASIS IDE to create a copy of the standard report in the "prog" directory of our project area (Figure 8a), and then add code to open and retrieve data from our Vendor Categories table.

Our custom report shown in Figure 9 is now totally contained within our custom project area, so we'll be able to re-incorporate it after our next upgrade.

Step 5: Re-install Customizations After an Upgrade

Barista's ability to facilitate this sort of customization adds value to the product and also ensures that the product isn't frozen in time! All of the modifications to the base product are preserved in our project area so we can upgrade the base product and then use Barista's Install Application wizard to re-incorporate our customizations (Figure 10).

After an upgrade, simply point the Install Application wizard to our project area. Barista analyzes the resource files in the project (`<project>/data/bar/*.xml` files) and compares information in those files with the current dictionary to check for possible conflicts. If it finds issues or conflicts, you can print a report that lists them and decide if the issue level should be lowered (i.e., no longer considered a critical issue) or if you need to make changes in the project in order to re-synchronize it into Barista. The latter may occur if, for example, you added a new field as part of your customizations, and that new field has now also been added to the upgraded product.

When no impeding issues remain, Barista processes the project's resource files, incorporating their data back into the current dictionary, then rebuilds any tables, forms and callpoints as necessary. When the wizard is complete, your modifications are once again part of the base product.

Conclusion

The Barista Application Framework gives you the tools you need to provide customized solutions that can be preserved through the upgrade cycle. Now you can help the user address their unique business needs without the fear of becoming frozen in time. So … let it snow! ■

By **Jeff Ash**
Software Engineer

Database Update Wizard

A very common problem for application developers to consider is how to update an older installation so that it is compatible with a new version of the software. If it were a simple application such as a word processor or spreadsheet application, they could just install the new application files and everything is ready to go. However, with database-driven applications, this is not usually enough; developers must also update the database to hold new data required by the new version and the addition of new tables to the database. To further complicate things, the database used with the old version may have customizations that make it difficult to simply copy data from the old tables to the new ones.

Using the new Database Update Wizard, developers can save time and energy creating an update solution for their applications. The wizard takes a source database, new data dictionary, and destination for the new data files, and builds an online copy job that copies the contents of the existing database into data files that match the definition of the new dictionary. The administrator has the opportunity to propagate any modifications in the old dictionary that are not present in the new dictionary.

Using the Wizard

The Database Update Wizard, located in the BBj® Enterprise Manager, uses a very simple interface. To access the wizard, select the source database (your existing application's database) from the list of databases in the Enterprise Manager. On the database "Information" tab, click the [Update Wizard] button at the bottom to launch the wizard.

The first panel of the wizard prompts for an optional database name, destination dictionary, and location for the newly created or modified data files.

1. If specifying a database name, a new database is added to the Enterprise Manager that points to the new dictionary files.
2. Specify the
   a. Location of the new data dictionary that is defined for the new application.
   b. Location in which the new data files will be created that match the new dictionary definition.

```
declare com.basis.api.admin.BBjAdminBase adminBase!
declare com.basis.api.admin.BBjAdminList adminList!

dbserver$ = "localhost"
dbport = 2002
dbssl = 0
user$ = "admin"
password$ = "admin123"

adminBase! = com.basis.api.admin.BBjAdminFactory.getBBjAdmin(
:  java.net.InetAddress.getByName(rd_dbserver$),
:  dbport,
:  dbssl,
:  user$,
:  password$)

db! = adminBase!.getDatabase("ChileCompany")

REM Specify the location of the new dictionary and the destination
REM for the data files. The returned object will contain all the
REM info to perform an update job. It will contain all of the missing
REM database items, so you will probably want to clear that out or
REM remove items you do not want migrated. See below for that.
info! = db!.getDictionaryMigrationInfo(
:  "C:\databases\chiledd-new\bbdict",
:  "C:\databases\chiledd-new\data")

REM Start the update process on the server.
db!.updateDatabase(info!)
```

Figure 1. Example of launching a simple update operation from a BBj application

The next panels show any tables, views, stored procedures, columns, etc. that are present in the old dictionary but not present in the new dictionary. This gives the administrator the opportunity to move over any modifications from the old database that still need to be available to the new application. Select those items that should be brought into the new database.

When the wizard is complete, an online copy job begins and creates the new data files that match the new dictionary definition, and then copies the existing data into these new data files. To monitor the progress of the online copy job, select the "Online Copy Jobs" node in the Enterprise Manager navigator area. When the job reaches 100%, select it, and then click the [Finish] button to perform the final cleanup and closing of files.

But Wait, There's More!

Developers can also perform a database update operation programmatically using the new BBj® Admin API. The example in Figure 1 shows how to perform a simple update operation from a BBj application. Figure 2 shows the Online Copy Job in Enterprise Manager.

Summary

Using the Database Update Wizard or the equivalent Admin API calls, an administrator or developer can make the process of updating the structure of their database to a new structure less arduous. Administrators can perform this operation manually and monitor and select each piece as they go along. An application developer can initiate this operation without the need for any human interaction. ■

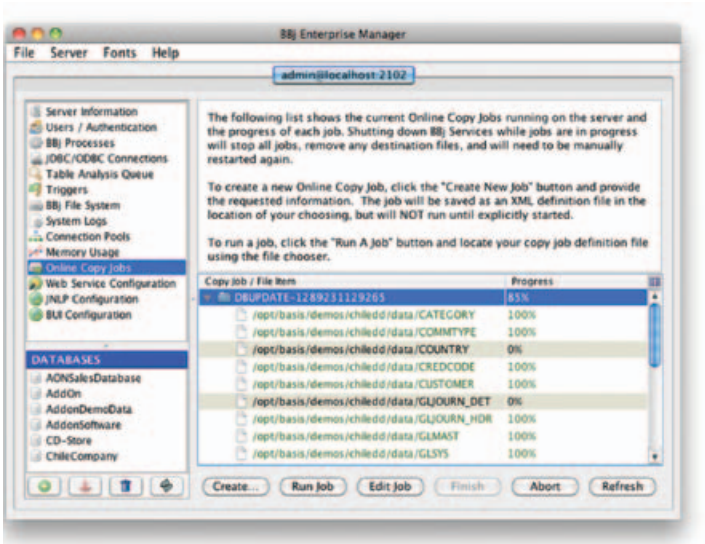


Figure 2. The Online Copy Job performing the Database Update

- Read the online documentation for Admin API Javadocs at links.basis.com/iocgg
- Try the sample at links.basis.com/dbcode

BBjGeolocation is Here!

The Global Positioning System, or GPS, technology has been booming lately. Its accuracy is even getting so precise that it is possible to pinpoint an exact location on this big planet of ours to within a few meters. This positioning capability is now built into many smart phones, wrist watches, cameras, automobiles, laptops, and, of course, dedicated GPS devices. The benefits and uses are varied and limited only by one's imagination.

A few of the more popular applications of the GPS technology deal with mapping, tracking movement, and providing directions. BASIS engineers get just as excited by these new technologies as everyone else, but when the feature provides practical use cases, they start thinking about ways to integrate those features into the product line.

By **Jim Douglas**
Software Engineer

Nick Decker
Engineering Supervisor

Since BBj is built on Java, many new capabilities are already within reach by accessing pre-built libraries. The capability to provide location information is already built into several browsers so wouldn't that be another great feature to add to BASIS' Browser User Interface (BUI)?

GPS or Geolocation?

While both GPS and Geolocation deal with positioning, Geolocation is the automatic detection of the geographic location of a device. Geolocation focuses on providing a meaningful location and attempts to supply more specifics by utilizing various sources for the information. GPS is just one of many technologies that Geolocation utilizes to provide location information. Geolocation returns the position as a 'latitude,longitude' pair along with an estimated accuracy value. For example, the location of the BASIS offices in Albuquerque reported 35.150036,-106.593957 with an accuracy of 30 meters. ▸▸▸

Continued at
links.basis.com/geolocation

By **Jeff Ash**
Software Engineer

LDAP/Active Directory Authentication in BBj

To help developers provide security for their applications and servers, BBj® provides a level of access control requiring a user name and password. While this is built in and very easy to use, it may make the job of system administrators easier if they can use existing user authentication mechanisms already available on their network. A simple BBj configuration change allows the use of other authentication mechanisms such as Microsoft's Active Directory.

Using Active Directory, administrators do not need to maintain multiple lists of user accounts, but rather, manage all user accounts from a single authentication source. This article takes a look at the two parts in the configuration that allow BBj to use Active Directory for authentication: the Active Directory server and the BBj Services installation.

What to do on the Active Directory Server

As a matter of convention, this article describes a convenient method for setting up the Active Directory server to interact with BBj properly. Follow these steps to ensure proper setup:

1. Using the ADSI Edit tool on the Active Directory server, create a directory structure as shown in Figure 1. The "Basis" and "UserPermissions" items should be of class type "container".
2. Next, add an object by right-clicking and then selecting "New Object" for the primary user account that will be used to administer BBj Services. Our example uses "admin" but you may choose "administrator", "jdoe", etc.; the object type should be "person".
3. Right click on the new user object and select "properties". Double-click the "description" attribute and enter "ALLOW_ALL" in the text field type, then click [Add] and [OK]. This grants all BBj permissions to this user for managing other users using the Enterprise Manager.
4. Right click on the BASIS object that we created earlier and select "Properties" and set the object's security to read/write for everyone. Repeat for the UserPermissions object. This allows BBj to update user information automatically rather than doing it manually for each user.

BBj Configuration

The BBj configuration involves two parts; the server information, and query information. The server information tells BBj how to connect to the Active Directory server to look up users and their permissions. The second part tells BBj how to find the information it needs once it is connected to the server.

To configure the Active Directory information, click the "Users/Authentication" item in the Enterprise Manager navigator and then select the "Authentication Settings" tab shown in Figure 2.

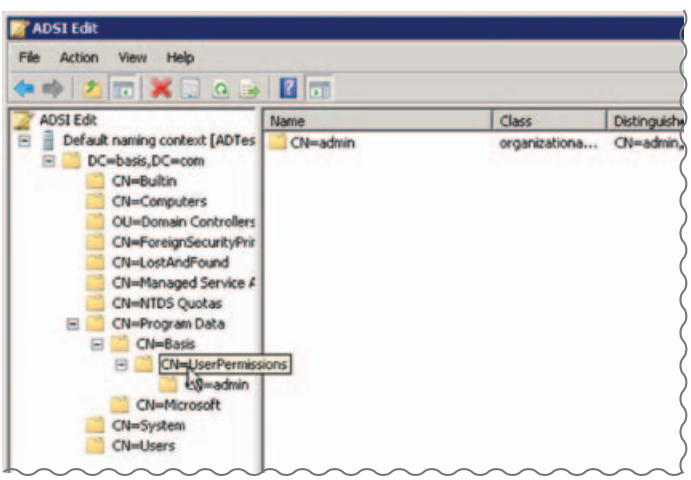


Figure 1. Create a sub directory structure with class type "container"

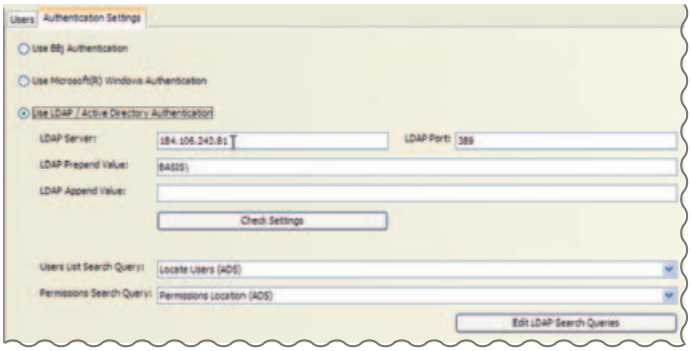


Figure 2. Active Directory Authentication configuration in the Enterprise Manager

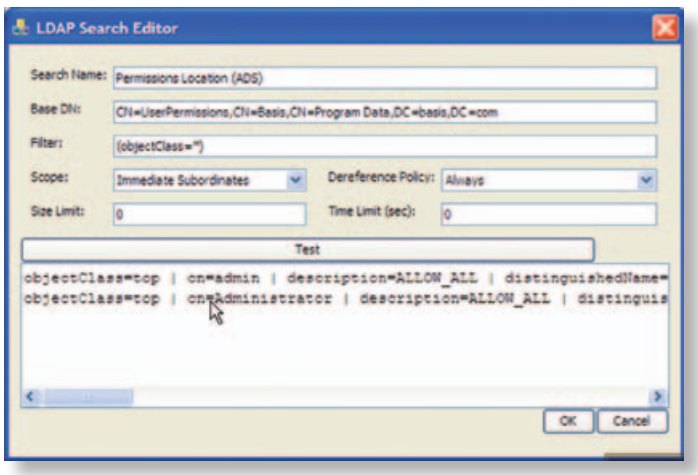


Figure 3. Sample text to enter in LDAP Search Queries

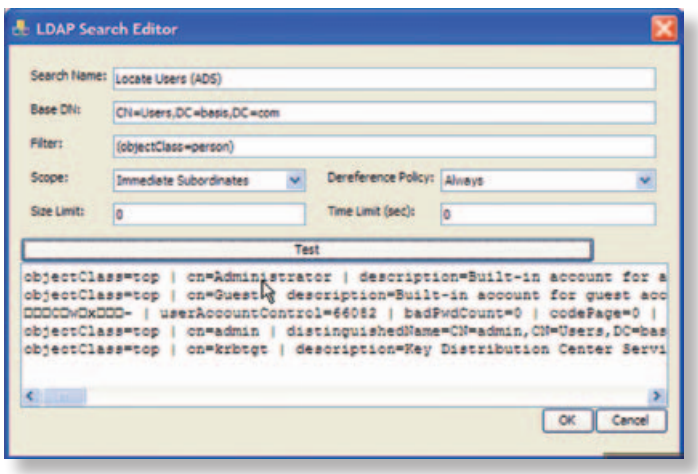


Figure 4. Data to test the query

Server Information

To configure the LDAP section of "Use LDAP/Active Directory Authentication,"

1. **LDAP Server**: Enter the IP address or host name for the Active Directory server.
2. **LDAP Port**: Enter the port number. The default port is 389 unless this was intentionally changed by the system administrator.
3. **LDAP Prepend Value**: Enter the domain of your Active Directory server. This is the string prepended to your user names when you log in to the Active Directory box. Our example uses `basis\`. This value prepends to all of the user names when sent through for authentication.
4. **LDAP Append Value**: Leave the field empty and click [Check Settings] to verify that all of the information is correct. You can authenticate using the user for whom you create an entry under UserPermissions in the ADSI Edit tool.

User List and Permissions Search Queries

The Query section of "Use LDAP/Active Directory Authentication" can be a little trickier. You will provide two valid LDAP syntax search queries, one for looking up users, and the other for looking up user permissions.

1. Click [Edit LDAP Search Queries] button.
2. Click [Add].
3. Enter the information shown in Figure 3 and click [Test] button to see the results of the query.
4. When satisfied with the results, click [OK] to save the query.
5. Click [Add] again.
6. Enter the information shown in Figure 4 and click [Test] to check the results
7. Finally, click "OK" to save this query.

Now that the queries are created, select the "Locate Users" query for the "User List Search Query" dropdown, and select the "Permissions Location" query for the "Permissions Search Query" dropdown.

Configuration is now complete and ready to save. Click the save button at the bottom of the panel to save these changes. A login dialog will appear which will force a relogin to the system using the user account we setup in the Active Directory server UserPermissions section earlier.

If you make any mistakes during the configuration process, it may be necessary to manually change the BBj.properties file in order to get logged back into the Enterprise Manager. If this is necessary, open the BBj.properties file in your favorite text editor and make the following property change which will set everything back to using BBj authentication, but will not remove any of the Active Directory configuration setting put in place (make sure to restart BBj Services after making the change to the property). You can then go fix the issue and try again:

```
com.basis.auth.type=multiserver
```

Conclusion

Using Active Directory authentication can make the job of the system administrator much easier by eliminating the need to maintain multiple sources of users and passwords. If your organization does not use Active Directory, then it would probably not be worth the effort to only use it for BBj authentication. However, if Active Directory is already in use, it makes sense to take advantage of this feature of the BBj authentication system to simplify the initial installation and reduce the ongoing maintenance of replicating of the same or similar authentication information throughout the organization. ■

By **Brian Hipple**
Quality Assurance Supervisor

Party Time – LaunchDock Goes Pure Java

If you have recently attended a BASIS TechCon or run a BASIS demo, then you are likely familiar with the slick BBj® LaunchDock application building-block utility shown in Figure 1. Amongst other functionality, the LaunchDock utility provides a cool animated visual feedback response when clicking a menu icon.

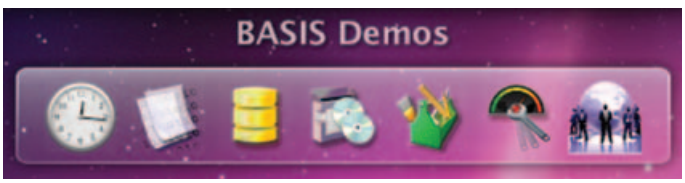


Figure 1. The BASIS Demos LaunchDock

The LaunchDock utility houses and runs the demos and in BBj 8.0, became a shipping BASIS utility able to launch any BBj application, operating system executable, document, or Web page. A simple right-click gives the user control over whether it is docked to one of the four edges of the screen and whether to make it "always-on-top". What you probably didn't know was that it depended on a combination of client-side Java objects and third party native C-code to show the translucent window (partially transparent, or see-through window).

In BBj 10.0 and higher, this is no longer the case. Read on to learn about the new integrated solution and what it means to you.

Third Party is NO Party

Depending on this third party code presented a couple of issues. First, the code on which LaunchDock depended was a shareware project that was never officially released. Second, the native code was only available on a few platforms that BASIS supports: 32-bit Windows, 32-bit Linux, and the Mac. Anyone running on other platforms could not use LaunchDock. Even though we did not want to depend on a third party product, Java did not support transparent windows, so we had no other choice.

Pure Java IS a Party

Beginning with release 1.6 update 10, Java supports transparency. Since BBj 10.0 requires version 1.6, we changed the code that generates the transparent windows in the LaunchDock utility to use the new `com.sun.awt.AWTUtilities` package instead of the third party C-code. This package also includes the ability to create irregular or non-rectangular shaped windows. The LaunchDock code serves as a working example for these features and gives developers a head start on creating their own transparent and shaped windows.

So what does that mean? Now anyone can run the LaunchDock demos on any platform that has an Oracle JVM (formerly from Sun). More importantly, you can use the utility to run your own BBj application code, just as BASIS does with its own production system (see Figure 2). It is slick and easy and fun.

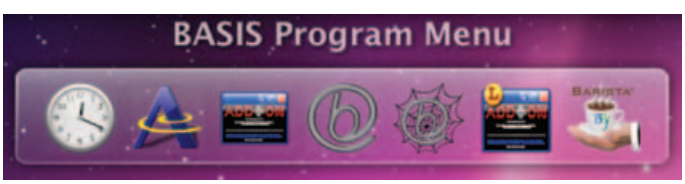


Figure 2. The BASIS Program Menu LaunchDock

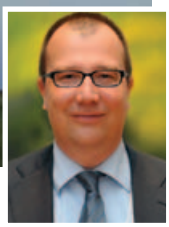
The Party Scoop

Using third party code in your application can be a smart way to bring new technology to the market while not having to wait for a release of your base product to implement new and important functionality. However, if you have included this technology into your base product, now is an ideal time to consider switching your program menu launcher to this integrated solution.

Just as we have found with our new LaunchDock, reducing dependencies on third party components greatly facilitates improved functionality and compatibility and opens up doors for a higher level of support. And for some who might not have been able to consider the third party providers due to platform limitations, this Java-based solution is platform independent and runs on 64-bit operating systems.

Likewise, you no longer have to rely on third party suppliers for such capabilities as forms, PDF, fax, e-mail. They too, once upon a time, depended on third party libraries but are now available directly in BASIS technology. If you are not using the power and versatility of the BASIS language for these capabilities, now is a good time to include them and begin enjoying the benefits. We would like to hear about your success with these components, including the LaunchDock, or how you are using transparency or shaped windows within your application. Let the real party begin! ■

- For more information about transparency, read How to Create Translucent and Shaped Windows links.basis.com/ebjgh
- Learn about these integrated BASIS solutions
  - Forms –
    - BBj Form Printing links.basis.com/bbjforms
    - Dishing up Forms links.basis.com/forms
  - Forms/PDF – PDF Now Also Means Perfectly Displayed Forms links.basis.com/pdf
  - PDF – BBj PDFs - Perfectly Displayed Formats on page 17 in this issue
  - Fax and e-mail – run the demos under "Demo LaunchDock" included in the BBj download

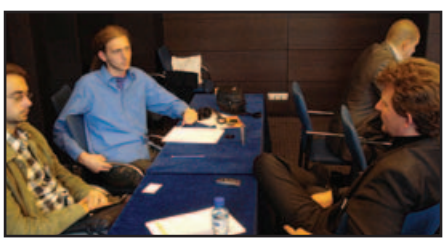


By **Patrick Schnur**
Marketing / Public Relations

BASIS Successfully Prospects in Bulgaria

An expert audience follows the AddonSoftware presentation in Sofia. All IT professionals in Bulgaria understand and speak English, the global lingua franca of the industry.

Chief Technology Officers and programmers from Bulgarian IT companies follow Stephan Wald's presentation in Sofia.



During the event, Stephan was interviewed by Alexander Glavchev and Tihomir Ivanov, editors of Bulgaria's leading IT magazine "Computerworld," about BASIS' plans in the region.

Bulgaria, located on the Black Sea with 7.5 million inhabitants, has a long standing tradition of excellence in mathematics and computer science (and chess!), dating as far back as the pre-Communist era. In 2002, the country earned an 8th place ranking in the world by total number of ICT specialists, out performing countries with far larger populations. Today, Bulgaria has an industrialized, open free-market economy with a large, moderately advanced private sector. The World Bank classifies it as an "upper-middle-income economy."

A member of the European Union since 2007, Bulgaria has experienced rapid economic growth in recent years. Though the country is momentarily struggling with the effects of the recent banking crisis, it has great growth potential for the years to come, especially since many software companies use Sofia as a hub to access neighboring countries on the Balkan peninsula and in the ex-Soviet Union.

BASIS Europe enjoys many loyal customers throughout most of the Western European countries, however the former Warsaw treaty nations are still uncharted lands for BASIS. As part of our strategy to explore market possibilities in these countries that were once behind the iron curtain, Stephan Wald and I traveled this past September to Sofia, the capital of Bulgaria.

Stephan Wald, BASIS Europe Manager, presented the AddonSoftware concept to the representatives of 20 Bulgarian ISVs, most of whom are interested in becoming AddonSoftware resellers. They seek an alternative feature-rich and customizable ERP solution as the products that currently dominate the market such as SAP and MS Dynamics are too complex and expensive for most mid-market companies.

The ISVs were very enthusiastic and liked the idea of ERP building blocks they can modify and customize themselves, giving them the opportunity to adopt easily to the needs of different industries in their country. Computerworld, Bulgaria's leading computer magazine, extended this ground breaking exposure by publishing an interview with Stephan about BASIS' plans in the region.

Encouraged by the tremendous interest we encountered in Bulgaria, we will put out our feelers to Poland and Romania next. Stay tuned!

By **Bruce Gardner**
Technical Support Supervisor

Virtual Licensing

Key to Apps in the Cloud

t's no secret that many organizations are going virtual. Virtualization is fast becoming the go-to technology for providing cost-effective disaster recovery, load balancing, and reliable service.

This past Spring, BASIS International Ltd. made the leap into the clouds – moving all production servers, license servers, build servers, Bugzilla servers, and even some test servers to Amazon's Elastic Compute Cloud (Amazon EC2). While planning the move to the cloud, our engineers put the finishing touches on a licensing feature to make it easier for customers to make the same leap: Virtual Licensing.

Virtualization is software that allows a piece of hardware to run multiple operating system images at once. The 'host system' hardware is detached and virtualized for each of the guest operating systems. While BBj® and (V)PRO/5 (Visual PRO/5® or PRO/5®) are well suited to such an environment, hardware virtualization has traditionally been a problem for the BASIS License Manager (BLM). To prevent software theft, every BASIS license is generated with a composite host ID based upon the hardware profile of the system running the BLM. When a virtual machine reboots, the guest operating system may come up with a different hardware profile than it had before, triggering 'nag' messages from the BLM. For this reason, in the days prior to version 10, BASIS recommended installing the BLM on a non-virtualized machine.

BASIS introduced virtual licensing in BBj and (V)PRO/5 10.0, making it possible for the BLM to work happily next to BBj or (V)PRO/5 on the guest operating system of a virtual server in the cloud or under your desk.

Here's how it works: The BLM automatically re-registers for a new license from our license database server each time the virtual BLM machine starts up with a different hostid. The BLM also automatically re-registers a license when it is within 24 hours of its expiration; once per hour for the license life until it successfully gets a new 24-hour license.

The FAQs

Q. What are the requirements for a virtual license?
A. To successfully run a virtual license, you must have the following:
- BLM 10.0 available in the BBj 10.0 release or downloaded separately with (V)PRO/5 10 (the new BLM can serve virtual licenses for new and old versions of BASIS products)
- An active Software Asset Management (SAM) subscription
- Internet access for the BLM to contact the BASIS license server (port 80)

Q. How do I purchase or upgrade to a virtual license?
A. Here are the paths to follow:
- Existing license(s) with an active SAM subscription: Contact the BASIS Customer Service Department at 1.800.426.5543 or cust@basis.com
- New virtual license(s): Contact the BASIS Sales Department at 1.800.423.1394 or info@basis.com

Q. How do I install and use a virtual license?
A. The process for installing and using a virtual license is exactly the same as a regular license but with the added requirement that the server on which the BLM is running must have access to the Internet.

- For more information, read the online BASIS documentation links.basis.com/vsjzz
- Check out these additional resources:
  - Amazon EC2 aws.amazon.com/ec2
  - Rackspace www.rackspace.com
  - VMWare www.vmware.com/products/server
  - Xen www.xen.org

By **Jeff Ash**
Software Engineer

Free at Last

Enterprise Manager Functionality

The BBj® Enterprise Manager (EM) is a powerful application that provides a user interface for performing various administrative tasks on a BBj Services installation and its databases.

In BBj 10.0, BASIS introduces a new Admin API that allows application developers to access the powerful features found in the EM, programmatically, from their BBj applications. Furthermore, an application can perform these administrative tasks on the server to which it is connected or another server on the same network.

What Can I Do?

You might be asking, "Sounds great, but what would be some of the things I might want to do"? Here are a few examples of tasks an application can perform without user interaction or by using a different application interface:

- Add/remove user accounts to the BBj system
- Create/remove a database
- Determine who has exclusive access to a record or file
- List and/or terminate running BBj processes or ODBC/JDBC database connections
- Start, abort, finish, or monitor online copy jobs for backup purposes or large file structure updates
- Start, abort, or monitor table analysis jobs
- Change database configuration settings
- Change server configuration settings
- Perform a database update/merge operation found also in the powerful new Database Update Wizard in the EM (see page 36 in this issue)
- Much, much, more!

How Can I Access This Functionality?

Accessing the Admin API functionality is very simple from any BBj application. The code in Figure 1 demonstrates how an application gains access to the Admin API.

The adminBase! object in this example provides the top level of access to the Admin API. Using the Javadoc documents found in the online BBj documentation under Language > BBjAPI > Java APIs, a developer can locate the methods and classes they need to perform their necessary operations. To help get developers started, the code in Figure 2 continues from the previous example and demonstrates how to create an online copy job, programmatically.

```
REM Use declare statements so that the BASIS IDE can help you using
REM its powerful code completion.
declare com.basis.api.admin.BBjAdminBase adminBase!
declare com.basis.api.admin.BBjAdminList adminList!

dbserver$ = "localhost"
dbport = 2002
dbssl = 0

REM Note that the user specified here must have permission to
REM perform the desired tasks. Check the EM if there is a question.
user$ = "admin"
password$ = "admin123"
adminBase! = com.basis.api.admin.BBjAdminFactory.getBBjAdmin(
:  java.net.InetAddress.getByName(dbserver$),
:  dbport,
:  dbssl,
:  user$,
:  password$)
```

Figure 1. Sample code of how an application gains access to the Admin API

```
REM Create the copy job
copyJob! = adminBase!.createOnlineCopyJobConfig()

REM Make a list to hold the list of files for the job
fileList! = copyJob!.getFileConfigs()

REM Add a file to the copy job
sourceFile$ = "C:/Program Files/basis/demos/chiledd/data/CUSTOMER"
destFile$ = "C:/Program Files/basis/demos/chiledd/data/CUSTOMER.bk"
fileConfig! = copyJob!.createFileConfig()
fileConfig!.setString(
: com.basis.api.admin.BBjAdminOnlineCopyJobFile.SOURCE_FILE,
: sourceFile$)
fileConfig!.setString(
: com.basis.api.admin.BBjAdminOnlineCopyJobFile.DEST_FILE,
: destFile$)
fileConfig!.setBoolean(
: com.basis.api.admin.BBjAdminOnlineCopyJobFile.AUTO_RECORD_SIZE, 1)
fileList!.add(fileConfig!)

REM Set the list of files to be copied
copyJob!.setFileConfigs(fileList!)

REM Start the job
adminBase!.startOnlineCopyJob(copyJob!)
```

Figure 2. Sample that demonstrates how to create an online copy job programmatically

Summary

BASIS responded to the requests of their customers and as of version 10.0, BBj now has the ability to perform EM-related tasks, programmatically, from within any BBj application.

Using the powerful Admin API, developers can perform administrative tasks, server configuration, database updating and migration, file backups, and more, completely under their own control.

This feature puts one more powerful tool into the hands of developers to make their applications even more robust and easy for their programmers to manage. ■

Try the sample at links.basis.com/emcode

See you in 2011!

CeBIT

March 1st to 5th
in Hannover, Germany

Viva BASIS TECHCON Las Vegas
May 9th-11th 2011

May 9th to 11th
in Las Vegas, USA

20 TechCon 11 EUROPE

June 7th to 9th
in Saarbrücken, Germany

BASIS International®

www.basis-europe.eu • www.basis.com