BASIS International Advantage

Volume 14 • Autumn 2010

The Dawning of a New Age with BUI Apps at EMQUE Page 6

TAI Scores a Whole in One With XCALL Page 8

A Quarter Century of Progression...the BBx Generations! Page 26

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

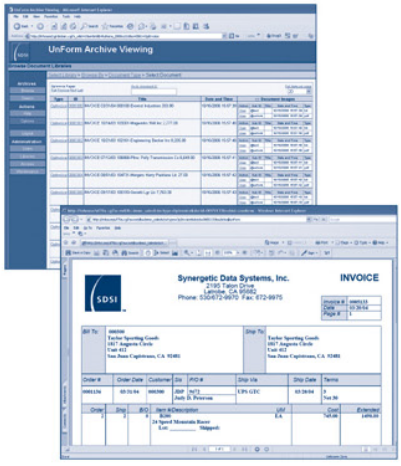
UnForm seamlessly integrates with any software application.

- 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
- Extensible 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

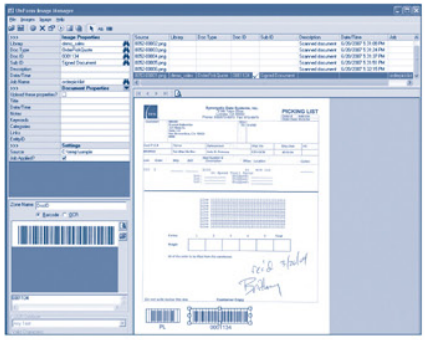


Universal web browser document retrieval

Document Imaging/Scanning

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

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



Document Image Manager

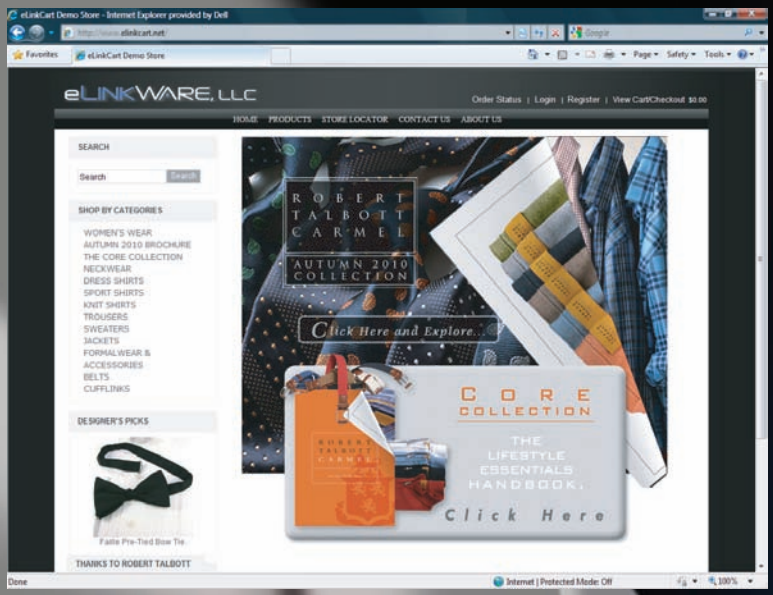
Synergetic Data Systems, Inc.
2195 Talon Drive
Latrobe, CA 95682 USA

(800) 446-7374 or (530) 672-9970
Fax: (530) 672-9975
sales@synergetic-data.com
www.synergetic-data.com

eLinkCart

The Complete eBusiness Solution

eLinkCart is the perfect business web portal to connect your business to the world wide web. The next step in e-commerce integration technology, eLinkCart gives you complete control in performing a wide variety of e-Business functions, including B2C and B2B ordering, customer self service, sales force automation, website analytics and web marketing. eLinkCart's unique building block design puts your future success in e-commerce at your fingertips.

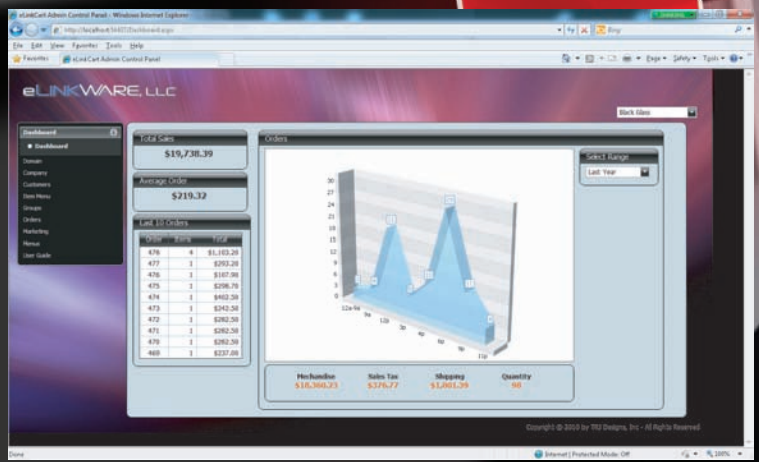


Customize your site's look and feel with one of our included themes, or create a completely customized look with our custom skin service.

Key Features:

- Integrated web content editor
- B2B & B2C e-Commerce, Customer Service Portal, Sales Force Automation and more
- Store locator with turn-by-turn directions
- Real-time shipping rate integration
- Multi-jurisdictional sales tax calculations
- Real-time payment processing
- Upsell and cross-sell capabilities increase average order size
- Customer Service portal allows B2B customers to view history, place orders, and make payments
- Built on ASP.NET framework

Can integrate with PRO/5 and BBj-based ERP applications!

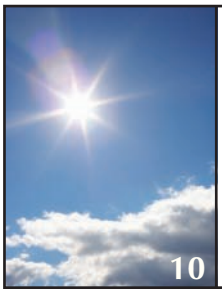


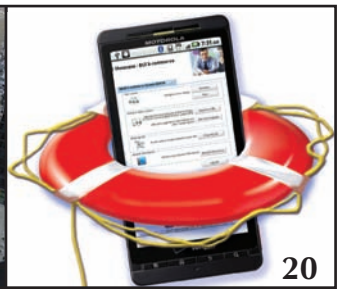
Web-based administrative control panel allows you to configure your website and monitor your traffic and sales 24/7 from anywhere.

Ordering directly from an interactive brochure allows you to group related products together and increase sales.

eLINKWARE, LLC
848 N Rainbow Blvd #1457
Las Vegas, NV 89107-1103

1-888-eLink-88 (1-888-354-6588)
info@elinkwarellc.com
http://www.elinkwarellc.com





Partnership

6 **The Dawning of a New Age with BUI Apps** – *By Susan Darling*
Discover the BUI revelation at EMQUE that is dramatically changing the look and feel of, and user's interaction with, their custom application.

8 **TAI Scores a Whole in One With XCALL** – *By Susan Darling*
See how this new verb lays a simple and solid bridge between a legacy Visual PRO/5 application and modern BBj functionality.

20 **BUI to the Rescue** – *By Laurence Guiney*
See one example of how to offer an application without requiring a JRE, while seeing a real-world example of BUI.

Language/Interpreter

12 **If it's Easier in BBj, Then Just XCALL it Forward** – *By Shaun Haney*
Follow along with this XCALL example to provide an easy no-fuss solution for creating a chart.

17 **BBjGeolocation is Here!** – *By Jim Douglas and Nick Decker*
Find out what this new feature is and how it might add value to your apps.

33 **BBj PDFs - Perfectly Displayed Formats** – *By Jason Foutz*
Deploy BASIS' PDF technology to create and distribute old or new reports in any of three easy methods.

46 **Enterprise Manager Functionality Free at Last** – By Jeff Ash
Access the familiar functionality in an all-new way and even automate its capabilities for the end user.

DBMS

36 **Database Update Wizard** – *By Jeff Ash*
Save time and energy creating an update solution for installations that require database changes.

45 **Automatic Database Analysis** – By Jeff Ash
Depend on this automatic database analyzer to dramatically boost the performance of SQL queries.

Development Tools

50 **BBj IDE in the Beginning - Then and Now** – *By Mike Phelps*
Take a closer look at today's integrated development environment and gain some insider tips.

System Administration

10 **Our Salvation is in the Cloud** – *By Dr. Kevin W. King*
Understand the whys and wherefores of moving to the cloud and gain confidence as you witness BASIS' transformation of their infrastructure in the cloud.

13 **Battle of the Browsers - BUI Wins** – By Nick Decker
Take a ring-side seat to discover why browsers matter and how you can end up on top with BUI.

Continued...

22 **Introducing the BASIS Custom Installer** – *By Brian Hipple*
Meet the BASIS Custom Installer; it is Pure Java and packs a lot of punch that will outlive its third party contenders.

25 **BBj's Web Integration is Better Than Ever** – *By Jason Foutz*
Discover new support for Secure Socket Layer connections for all types of Web connections.

38 **LDAP/Active Directory Authentication in BBj** – *By Jeff Ash*
Use this new feature to manage all user accounts from a single authentication source.

Building Blocks

18 **Plumbing the Barista Framework into BBj Forms** – *By Chris Hawkins*
Learn how to incorporate Barista menus and toolbuttons into your custom forms to extend functionality and provide a consistent look and feel.

28 **Preserving Your Customizations** – *By Chris Hawkins*
Have the best of both worlds - off the shelf functionality and customizations - that withstand the test of time and upgrades to the base application.

34 **AddonSoftware's Barista Building Blocks Strategy** – *By Paul Yeomans*
Understand the strategy upon which BASIS built a smart and savvy business solution.

40 **Party Time - LaunchDock Goes Pure Java** – *By Brian Hipple*
Reduce your dependency on third party components beginning with BASIS' own LaunchDock.

Columns

NEWS

41 **Mining the Future with Diamond 725** – *By Gale Robledo*

42 **BASIS Successfully Prospects in Bulgaria** – *By Patrick Schnur*

43 **Barista Gets a Hot Chile Reception in Santiago** – *By Jesús Alvarez*

44 **Canada's Software Showcase 2010** – *By Dave Foster*

44 **OSAS and BASIS in the Bright Lights of Las Vegas** – *By Gale Robledo*

TRZ

48 **Virtual Licensing Key to Apps in the Cloud** – *By Bruce Gardner*

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

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

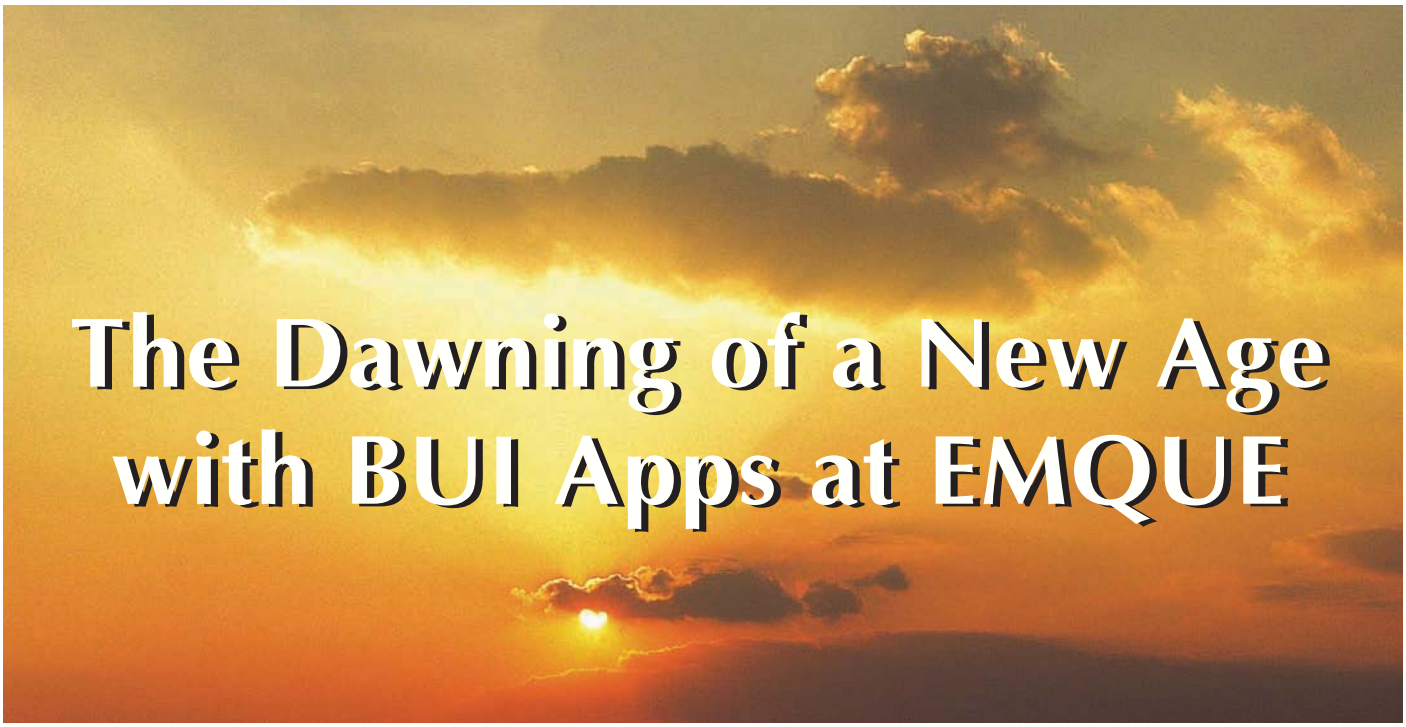
Electronic Production *Amer Child*
achild@basis.com

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

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

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

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



The Dawning of a New Age with BUI Apps at EMQUE

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.

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

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. >>

By Susan Darling
Technical/Marketing Writer

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.

TAI Scores a *Whole in One* With XCALL

A *whole* lotta power *in one* line of code

"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. >>

By Susan Darling
Technical/Marketing Writer

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.

Our Salvation is in the Cloud

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 7x24. 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.

By Dr. Kevin W. King
President & CIO

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.

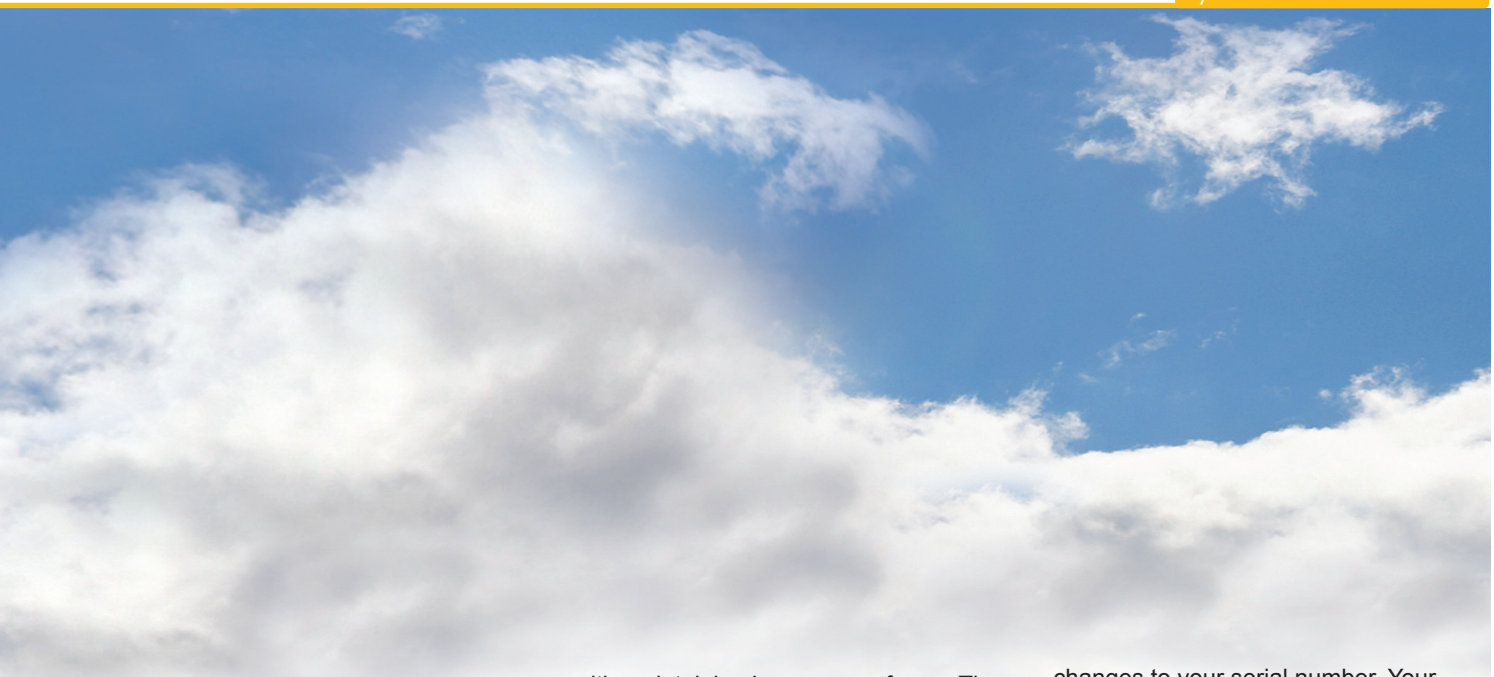
The Technology

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

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.

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. > >

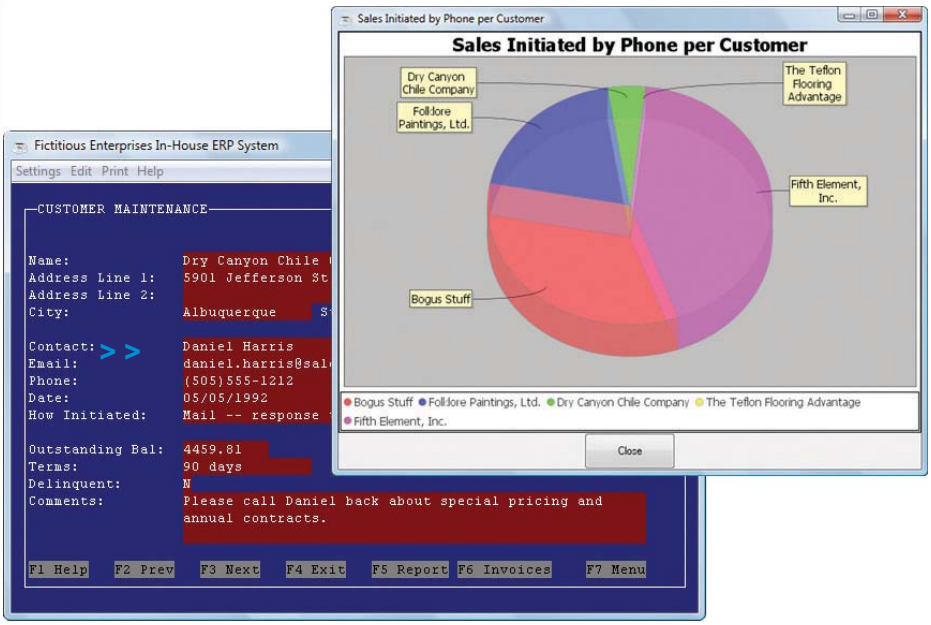


Figure 1. Sample pie chart from within the ERP application

By Shaun Haney
Quality Assurance Engineer

Planning

Having decided to use XCALL to show the sales people their pie charts from the ERP application, Ralph looked at how to configure (V)PRO/5 and BBj for XCALL. To accomplish this task, (V)PRO/5 needs to access BBj, and BBj in turn needs access to each salesperson's display.

BBj has some configuration options that are quite different from (V)PRO/5. (V)PRO/5 is mostly a single-tiered product which can use a data server, but the data server is optional. BBj is a client-server product with a central server to manage the interpreter, event-handling, file-handling, etc. when used in thin client mode, which is the default and most common configuration. All servers are part of BBj Services and while you can distribute BBj across multiple tiers it will run in a single-tier on the same machine without further configuration from the user.

The XCALL Server is a BBj program that handles (V)PRO/5's requests to run BBj subroutines and can run wherever it is needed. You can run XCALL Server centrally on the same machine as BBj Services if clients don't need a GUI or you can run an instance of the XCALL Server on each client's machine allowing BBj to display GUI as well as handle other kinds of requests.

Because Ralph wanted salespeople to be able to view the pie chart on their machine, he chose to run the XCALL Server individually on each machine. The PORTCOMMAND option, explained later in this article, allows (V)PRO/5 to launch the XCALL Server on demand, relieving the salesperson of any need to know that the XCALL Server is even there. Ralph installed the BBj ThinClient on a mapped drive, easily accessible from each salesperson's machine, saving him the trouble of installing and maintaining multiple installations.

Verifying (V)PRO/5 is Ready For XCALL

Beginning in version 10.01, (V)PRO/5 is preconfigured to run the XCALL Server in BBj 10.03 automatically when executing an XCALL statement. Ralph looked at his configuration to verify that (V)PRO/5 was configured correctly for his needs. First, he examined the short xcall.ini file to make sure the PORTCOMMAND option was in place. The PORTCOMMAND option automatically launches an instance of the XCALL Server on the same machine as (V)PRO/5. Then Ralph looked at the default line for PORTCOMMAND. On a Windows installation, the line reads: **> >**

```
portcommand="C:/Program Files/basis/bin/bbj.exe" "-cC:/Program Files/basis/cfg/config.min" -tT2 -q xcallserver.bbj - 0
```

Here's what this line contains:	
`C:/Program Files/basis/bin/bbj.exe`	Invokes the BBj interpreter
`-cC:/Program Files/basis/cfg/config.min`	Specifies the configuration file XCALL Server should use
`-tT2`	Specifies that the "invisible" SysGui terminal should be used. Users won't see the console, but will see GUI components of the application. In Ralph's case, the sales people will see the window containing the pie chart.
`-q`	Do not display the BBj splash screen when launching the XCALL Server.
`xcallserver.bbj`	By default, xcallserver.bbj is in BBj's PREFIX. Otherwise, it can be found at: **<bbj install path>/utils/XCALL/xcallserver.bbj.**
`- 0`	Specifies port 0 to the XCALL Server. When this port is specified, the XCALL Server will use whatever port is available.

Ralph made a couple of adjustments to the default PORTCOMMAND. Since he was accessing BBj on a mapped drive, he adjusted the paths to BBj and the config.min file. Then he added **-RHphydeaux** after the **-q** option, since BBj Services is running remotely on Phydeaux. Next, Ralph briefly perused (V)PRO/5's config.bbx file to verify it contains the line **XCALL=xcall.ini**, which indicates that (V)PRO/5 should load the xcall.ini file and that this file is in (V)PRO/5's current directory.

Writing the BBj Subroutine

Now that Ralph had (V)PRO/5 and BBj configured to use XCALL, it was time to write the BBj subroutine to display the pie chart. An XCALLed subroutine consists of an optional label, an ENTER statement specifying any arguments that the calling program will pass in, and an EXIT statement at the end of the routine, just like a (V)PRO/5 subroutine.

Ralph needed to decide what kind of data to pass in; XCALL supports strings, numbers, integers, and 1-, 2-, and 3-dimensional arrays. The BBjPieChart control takes a title as well as a name and a value for each slice in the pie so Ralph decided to pass the title "Sales Initiated by Phone per Customer" in as a string, the name of each customer in as a one-dimensional array of strings, and the number of phone-initiated sales for each customer as an array of numbers.

Ralph wrote a BBj subroutine that displays a pie chart based on the arguments passed in to the subroutine and then he saved this program on Phydeaux. He also added the pathname of the subroutine file to Phydeaux's prefix.

Calling the BBj Subroutine from (V)PRO/5

Once Ralph wrote the BBj subroutine, he decided to write a very short (V)PRO/5 program to test it before finally integrating the XCALL statement into the ERP application. He populated **title$** with the title of the chart and filled the **customer$** and **sales** arrays with string and numeric data, respectively. He then XCALLed the BBj subroutine with the following statement: **> >**

```
XCALL pathToBBjProgram$+"::show_pie_chart",err=handl_err,title$,customer$[ALL],sales[ALL]
```

Here's what this line contains:

pathToBBjProgram$	The variable Ralph used to store the name that the BBj program will call. The BBj program must be accessible to BBj Services, either by use of a full path on the same machine as BBj Services, accessible via a data server with use of data server syntax, or contained in BBj's PREFIX. (V)PRO/5's PREFIX does not affect BBj's being able to find the file.
"::show_pie_chart"	The label Ralph used for the BBj subroutine. XCALL can call subroutines without a label. However, using labels allows for including several BBj subroutines in the same file.
err=handl_err	The error branch for the XCALL statement, which like CALL, must be included before the subroutine's argument list. Including this expression after or within the argument list would cause it to be interpreted as a boolean comparison.
title$,customer$[ALL],sales[ALL]	The variables that comprise the argument list for the subroutine.

Once XCALL is invoked, (V)PRO/5 connects to the XCALL server, which then launches the BBj subroutine in its own BBj interpreter. It is this interpreter that will then show the window with the pie chart. Download and review the short (V)PRO/5 program, **xcall.bbx.src** and the subroutine Ralph wrote, **xcall.bbj.src**, from the link that appears at the end of this article.

Other Features of XCALL

This solution to Ralph's dilemma demonstrates a fairly straight-forward way to use the XCALL verb, but consider these additional XCALL verb command options:

- **Specify a timeout**
  The **tim=** option to XCALL allows the user to specify a timeout for XCALLing a program in number of seconds, including the amount of time it would take the BBj program to run and then respond back. This option is useful for non-interactive calls, where the network may be slow or unreliable; the default is 0 or "wait forever."

- **Employ the new error-handling options introduced with the XCALL verb**
  Error handling for XCALL introduces some new features. If an error occurs in an XCALLed program, XCALL will return an **!ERROR=85**. This is a general error informing the user that an unspecified error occurred in XCALL. The text for the error holds the real clue to what happened and can be retrieved in (V)PRO/5 by calling the **ERRMES** function with a value of -1. While in the past, **ERRMES(ERR)** gave a generalized message that corresponded to the error, **ERRMES(-1)** now gives specific information related to an error and is useful for any error, not just 85. Ralph decided to take full advantage of this new feature by printing out **ERRMES(-1)** as part of the diagnostics in his error branch.

- **Run the XCALL server locally or remotely and even on a different port**
  The host and port can be specified for XCALL in a couple of ways. The first way is to specify a host and port in an xcall.ini configuration file. The second way to specify a remote host and port is with the **mode=** option. A couple of the modes supported for XCALL are **"RemoteHost=<host>"**and **"RemotePort=<port>"**. By default, the values are **"RemoteHost=localhost"** and **"RemotePort=4444"**. Since Ralph is running **xcallserver.bbj** locally on each machine and not changing the port the XCALL Server runs on, he can accept the defaults.

- **Specify the PORTCOMMAND option at runtime**
  If needed, the PORTCOMMAND can be specified in the mode string with **"PORTCOMMAND=<command line>"**. This option is useful in situations where the location of the BBj ThinClient can not be determined before run time.

After Ralph tried out his XCALL statement in a sample program and verified that it worked, he proceeded to integrate the new pie chart feature into the sales force's software. Fictitious Enterprises now takes advantage of the new pie chart feature as well as many other BBj features through XCALL.

Conclusion

Fictitious Enterprises' fictional but feasible example shows one way to > >

use XCALL to leverage a BBj feature in (V)PRO/5. It also alludes to other ways to configure XCALL for non-GUI and non-interactive features. XCALL is a good and fast way to utilize features in BBj that are either difficult to implement or nonexistant in (V)PRO/5. Here are just a few other examples of other BBj features developers can access in (V)PRO/5 via XCALL:

- Fax documents
- Print PDFs
- Create VKEYED and XKEYED files
- Create BBjForms
- Create reports with iReport(r)
- Consume web services
- Parse XML documents
- Easily send email
- Public Key Cryptography
- Access databases via JDBC instead of ODBC
- Publish Google Docs

XCALL is a useful mechanism for transitioning code to BBj piecemeal, while still allowing a (V)PRO/5 infrastructure to continue running productively. The CALL-like syntax and the ability to transfer data via variables or access a common data file or DBMS, makes this transition a lot less code-intensive than an equivalent transition using SCALL and data files. Indeed, XCALL is handy within environments where BBj and (V)PRO/5 must coexist and interact. ■

- Download and extract links.basis.com/10xcallcode to run/review the sample subroutine xcall.bbj.src
- Re-take a Java Break with BASIS - "Mix 'n Match PRO/5 and BBj - REDUX" – for an example of XCALL. Click here and select "play." links.basis.com/javabreak
- Refer to the online documentation for more information on XCALL www.basis.com/onlinedocs/documentation/commands3/xcall_verb.htm

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.

By Nick Decker
Engineering Supervisor

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

Figure 1. The BBjListButton in different browsers and platforms

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.

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.

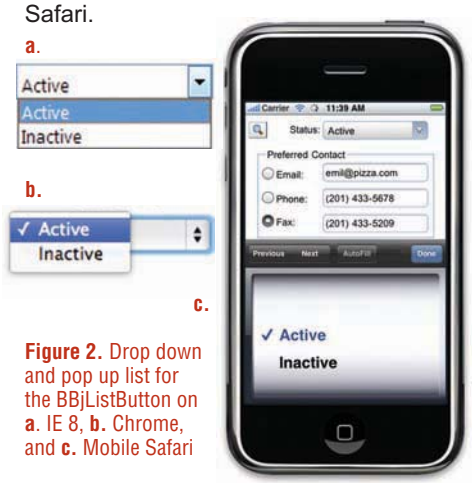


Figure 2. Drop down and pop up list for the BBjListButton on **a.** IE 8, **b.** Chrome, and **c.** Mobile Safari

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, 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 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 >>

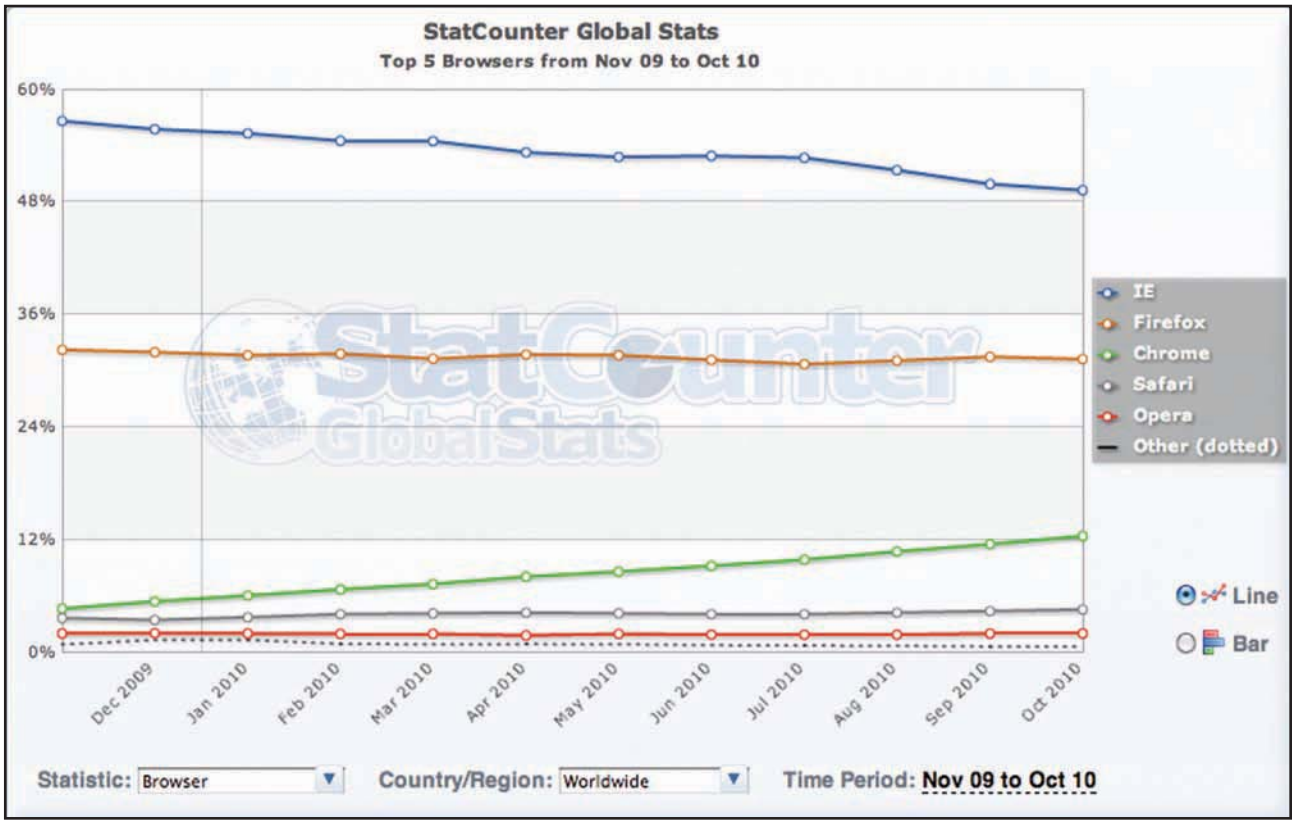


Figure 3. StatCounter's global browser share for the last year

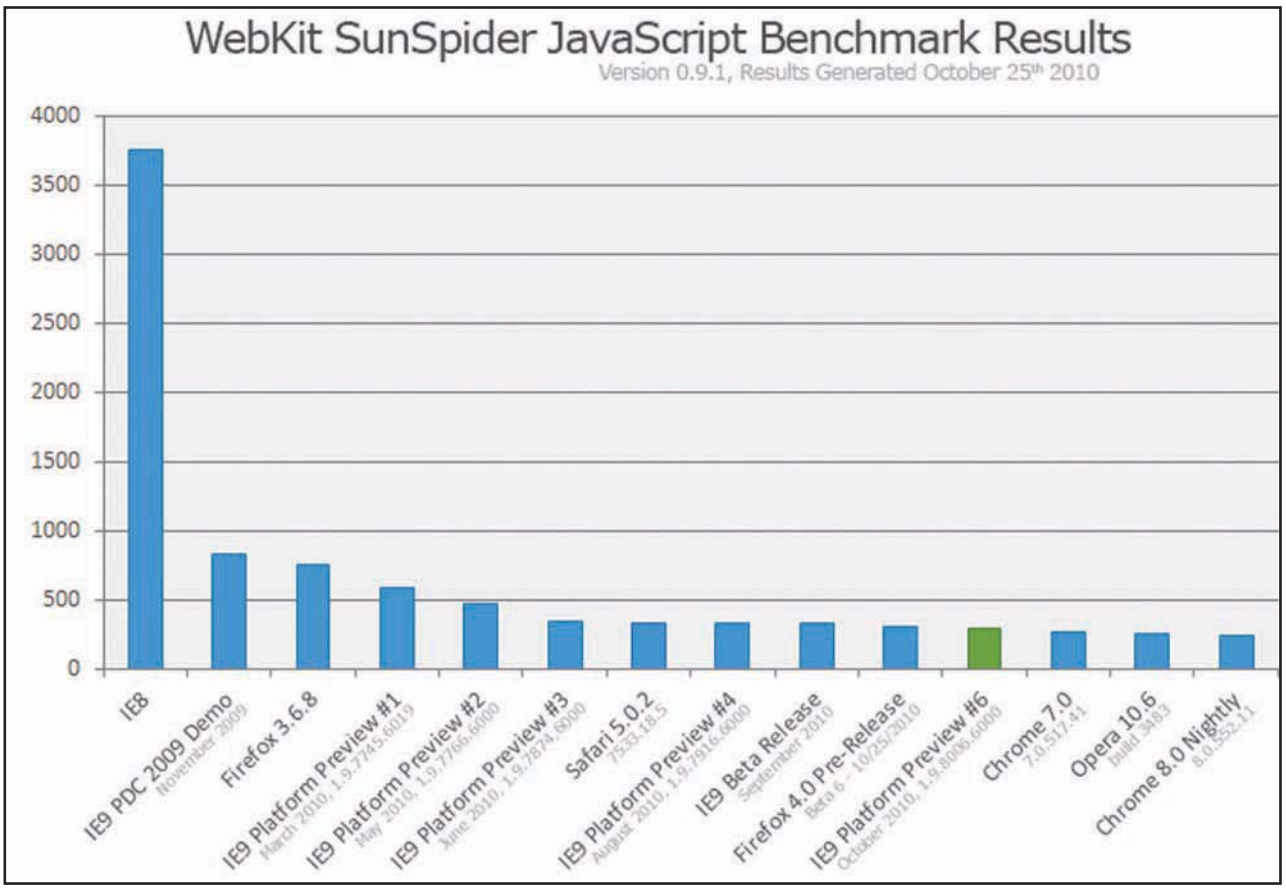


Figure 4. Benchmark results comparing various browser's JavaScript performance (lower is better)

>>

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 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.

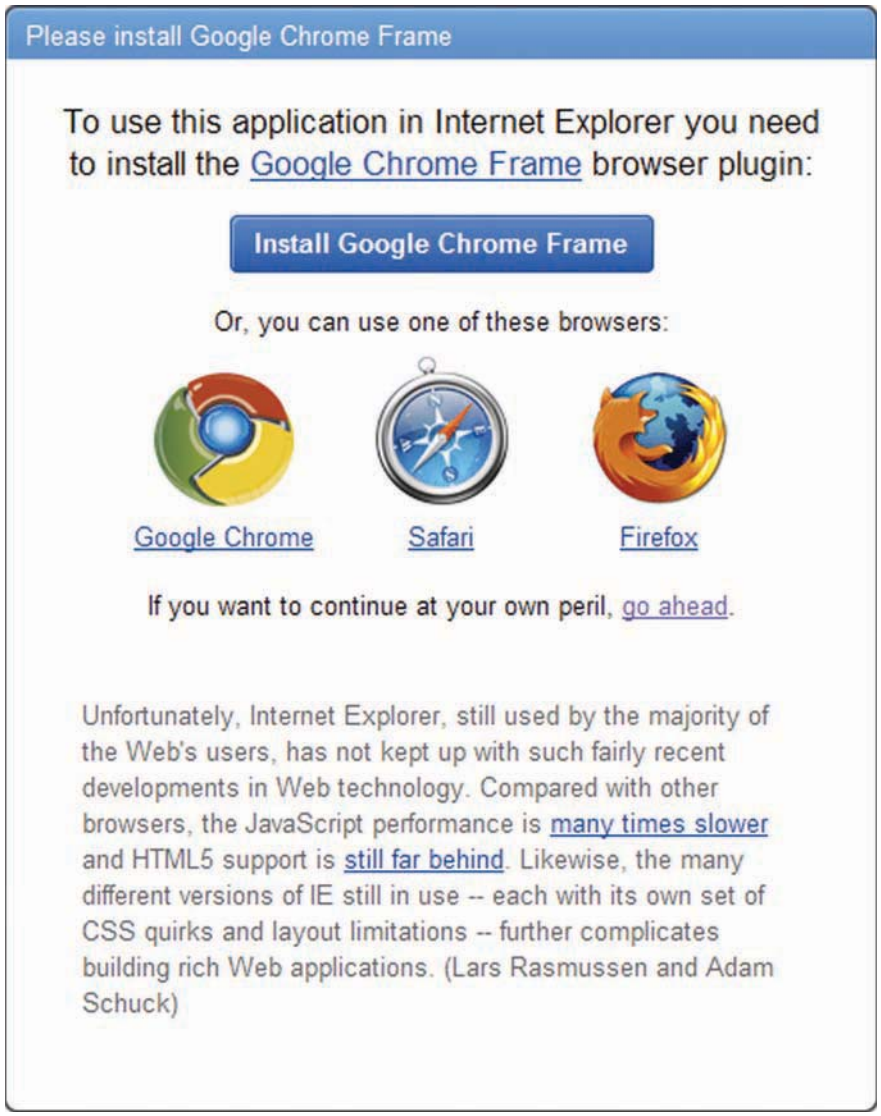


Figure 5. The screen prompting the user to install the Google Chrome Frame for IE

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.

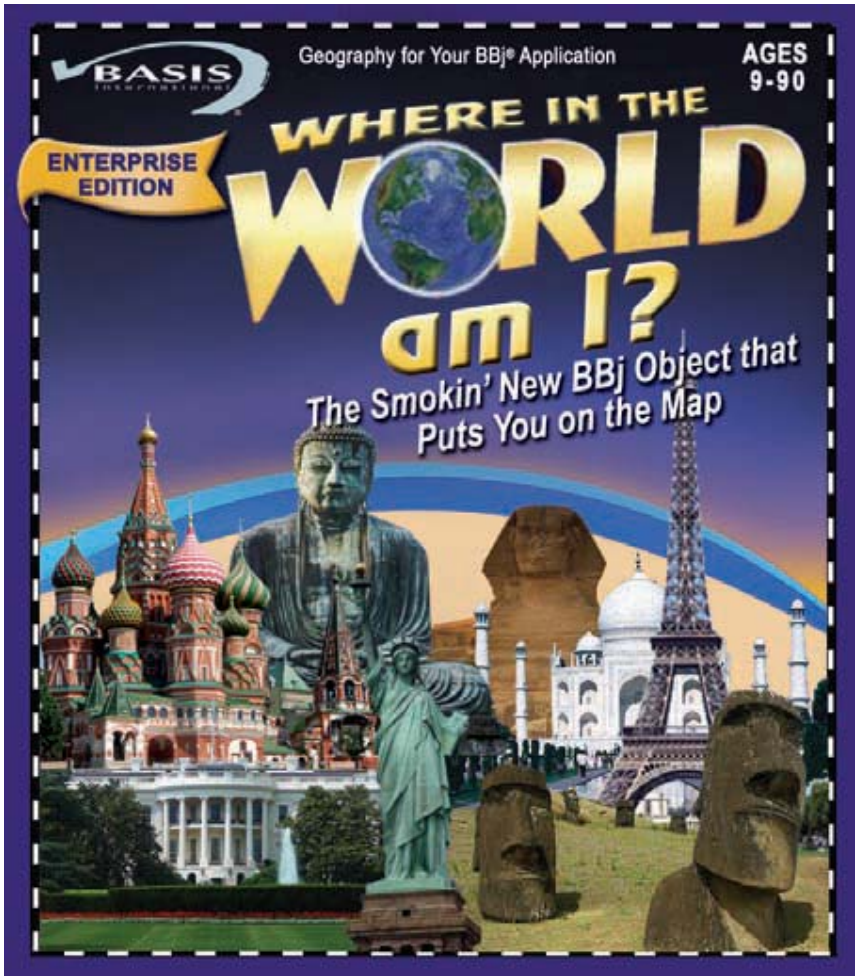
Summary

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.

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.



Since BBj is built on Java, many new capabilities are already within reach by accessing pre-built libraries, as has been the case for some of the recent additions such as e-mail and fax, PDF support, charts and graphs, Web Services, access to Google Docs, to name a few. 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, when possible, by utilizing various sources for the information. GPS is just one of many lower-level back-end 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. > >

By Jim Douglas
Software Engineer

Nick Decker
Engineering Supervisor

How it Works

Desktop and mobile browsers use a combination of methods to detect the client location. The most common methods are

- GPS - requires line of sight to the GPS satellites; accurate to about 15-25 meters; sometimes slow to establish the position
- Triangulation based on cell towers (accurate to ~1000 meters)
- Triangulation based on Wi-Fi access points (accurate to about 30 meters)
- A lookup of the registered geographic location of your IP address (extremely inaccurate, used only as a last-ditch fallback)

Desktop browsers don't usually have GPS; they typically use Wi-Fi Positioning System services from Google, Skyhook Wireless, and other providers.

Geolocation API

Most current browsers support a standard Geolocation API, which returns geolocation information in a common format, regardless of how the position is determined under the covers. The BBj 11.0 BBjGeolocation API (available as a preview feature beginning in BBj 10.02) uses this standard Geolocation API and provides an interface that closely mirrors the underlying browser API.

BBjGeolocation Sample

Because it can take a while to determine the geolocation position, requests are always handled asynchronously. That is, the program requests a geolocation position, and at some point in the future, an event is fired to return the requested information or an error code if the position could not be determined. The steps are as follows:

1. Get the geolocation object (this can throw an error if geolocation is unavailable):

```
geolocation! = bbjapi().getSysGui().getGeolocation(err=unavailable)
```

2. Define an event handler for the geolocation position event:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_POSITION,"position")
```

3. Request the current position:

```
geolocation!.getCurrentPosition()
```

The user must approve or deny this request, since determining a user's location without their express consent may potentially compromise their privacy. Therefore, programs are never allowed to retrieve the user's geographic position without the user's explicit permission.

The geolocation position (or an error code) is returned to this event handler:

```
position:
  position! = bbjapi().getSysGui().getLastEvent()
  if position!.getStatus() = 0 then
    latitude = position!.getLatitude()
    longitude = position!.getLongitude()
    accuracy = position!.getAccuracy(); rem ' metres
    rem ' do something with the position
  endif
return
```

The complete sample program is shown in **Figure 1** and is available to download and try at links.basis.com/10geocode. **> >**

```
rem ' Geolocation Position
   precision 6
   sysgui! = bbjapi().openSysGui("X0")
   title$ = "Geolocation Position"
   window! = sysgui!.addWindow(10,10,350,100,title$,$00090003$)
   window!.setCallback(window!.ON_CLOSE,"eoj")
   label$ = "Position (latitude,longitude):"
   label! = window!.addStaticText(100,25,25,320,25,label$,$$)
   status$ = "Waiting for position..."
   text! = window!.addStaticText(101,25,50,320,25,status$,$$)
   font! = sysgui!.makeFont("Courier New",12,sysgui!.PLAIN)
   label!.setFont(font!)
   text!.setFont(font!)
   gosub init

rem ' Process Events
   process_events

eoj:
   release

init:
   geo! = sysgui!.getGeolocation(err=unavailable)
   geo!.setTimeout(30)
   geo!.setCallback(geo!.ON_GEOLOCATION_POSITION,"position")
   geo!.getCurrentPosition()
return

unavailable:
   text!.setText(errmes(-1))
return

position:
   position! = sysgui!.getLastEvent()
   status = position!.getStatus()
   switch status
      case 0; rem ' success
         latitude = position!.getLatitude()
         longitude = position!.getLongitude()
         position$ = str(latitude)+","+str(longitude)
         text!.setText(position$)
         break
      case 1; rem ' user refused permission
         text!.setText("Failed (permission denied)")
         break
      case 2; rem ' couldn't calculate position
         text!.setText("Failed (position unavailable)")
         break
      case 3; rem ' 30-second timeout reached
         text!.setText("Failed (timeout)")
         break
      case default; rem ' unknown error
         text!.setText("Failed (status "+str(status)+")")
         break
   swend
return
```

Figure 1. BBjGeolocation sample

When a program wants to track the user's position, the browser must get permission from the user. **Figures 2a** and **2b** show how this looks in Chrome on Windows 7 and in Mobile Safari on iPhone, respectively. **> >**

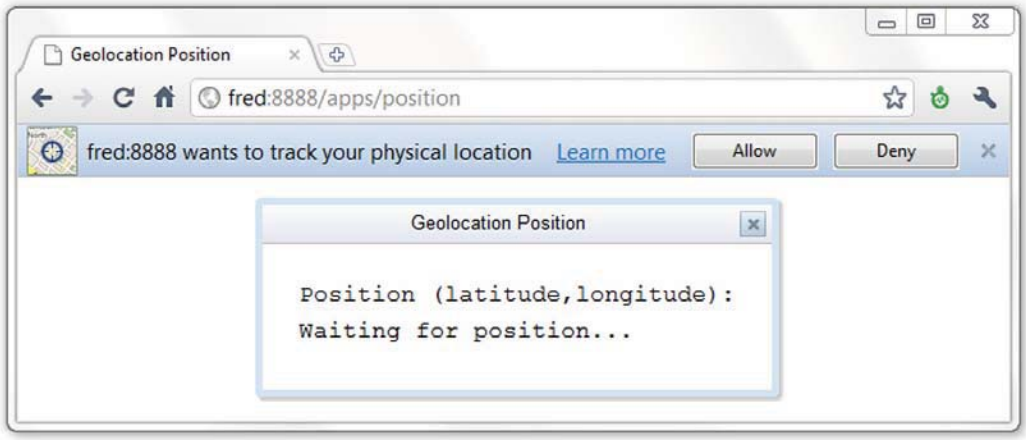


Figure 2a. Geolocation requesting permission (Chrome, Windows 7)

Figure 2b. Geolocation requesting permission (Mobile Safari, iPhone)

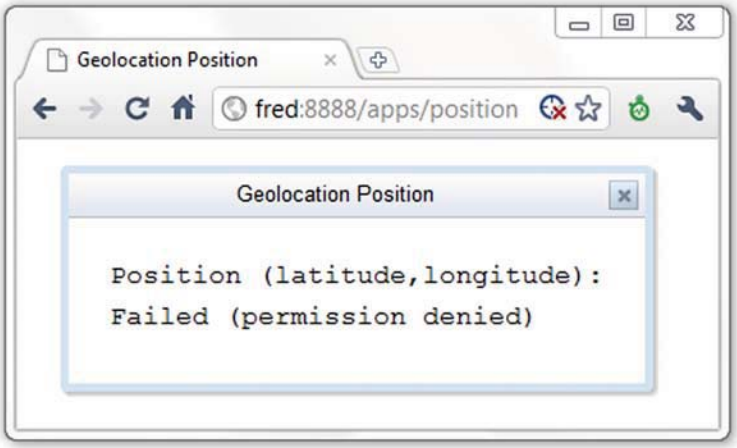


Figure 3. Geolocation reporting that the user refused permission

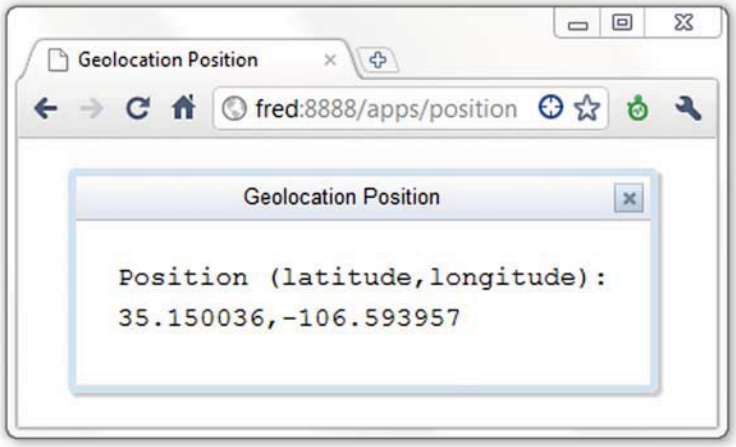


Figure 4. Geolocation reporting position

If the user denies the request or if an error or timeout occurs, an error status is returned in a BBjGeolocationEvent, as shown in **Figure 3**.

If the user allows the request, geolocation position information is returned in a BBjGeolocationEvent, as shown in **Figure 4**.

To verify the accuracy of the reported position, you can hand it off to Google Maps. For example, this shows the location of the BASIS office in Albuquerque:

http://maps.google.com/?q=35.150036,-106.593957

This sample requests a single geolocation position update with the following two lines of code:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_POSITION,"position")
geolocation!.getCurrentPosition()
```

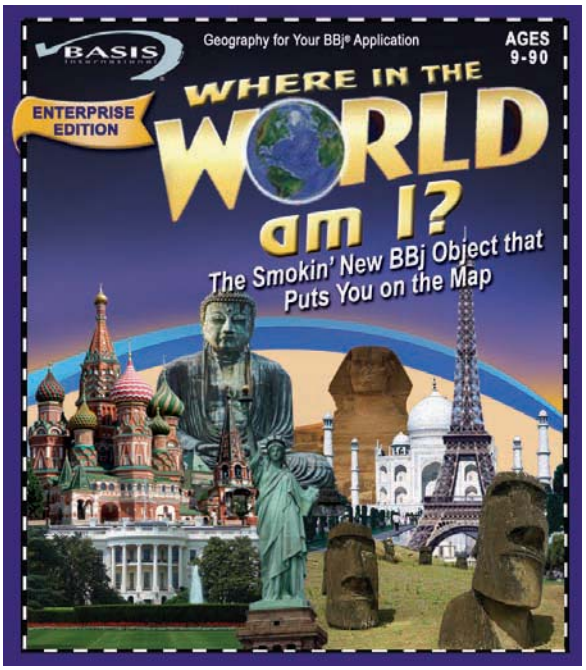
To receive regular position updates, substitute this single line of code in place of those two lines of code:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_WATCH,"position")
```

Summary

The proliferation of smart mobile devices with fully capable web browsers means that now your Business BASIC application can run almost anywhere at any time. By leveraging BASIS' BUI technology, your applications can run on dozens of mobile devices such as Apple's iPad and iPhone, Google Android devices, and dozens of others that come equipped with full-featured browsers. As BUI capabilities continue to expand, so do the potential features and abilities of your application. With the latest inclusion of Geolocation, you can now write a BBx program that manages real-time tracking of fleet vehicles, integrates with Google Maps, and provides other location-based services.

Give this latest feature a try and see how easy it is to answer at least the first part of the age-old question of "Where am I and why am I here?" ■



For more information about geolocation, read these information links:

- BBjGeolocation in the online BASIS documentation
- Location-Aware Browsing test page and FAQ from Mozilla
- How to use the W3C Geolocation API from Dev.Opera
- Geolocation from Dive Into HTML
- Geolocation API Specification from W3C

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.

By Chris Hawkins
Software Developer

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.

In the old form, as we began new data entry, we had specific code to disable the Update and

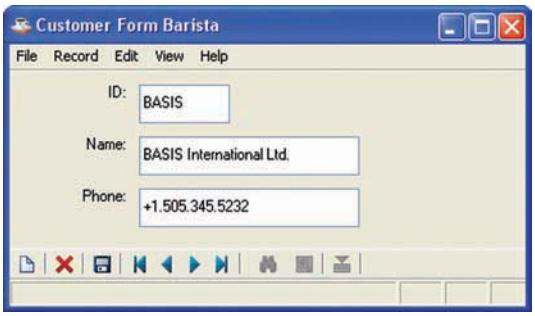


Figure 2. Customer form revised to include Barista menu/toolbuttons

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! ■

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

```
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

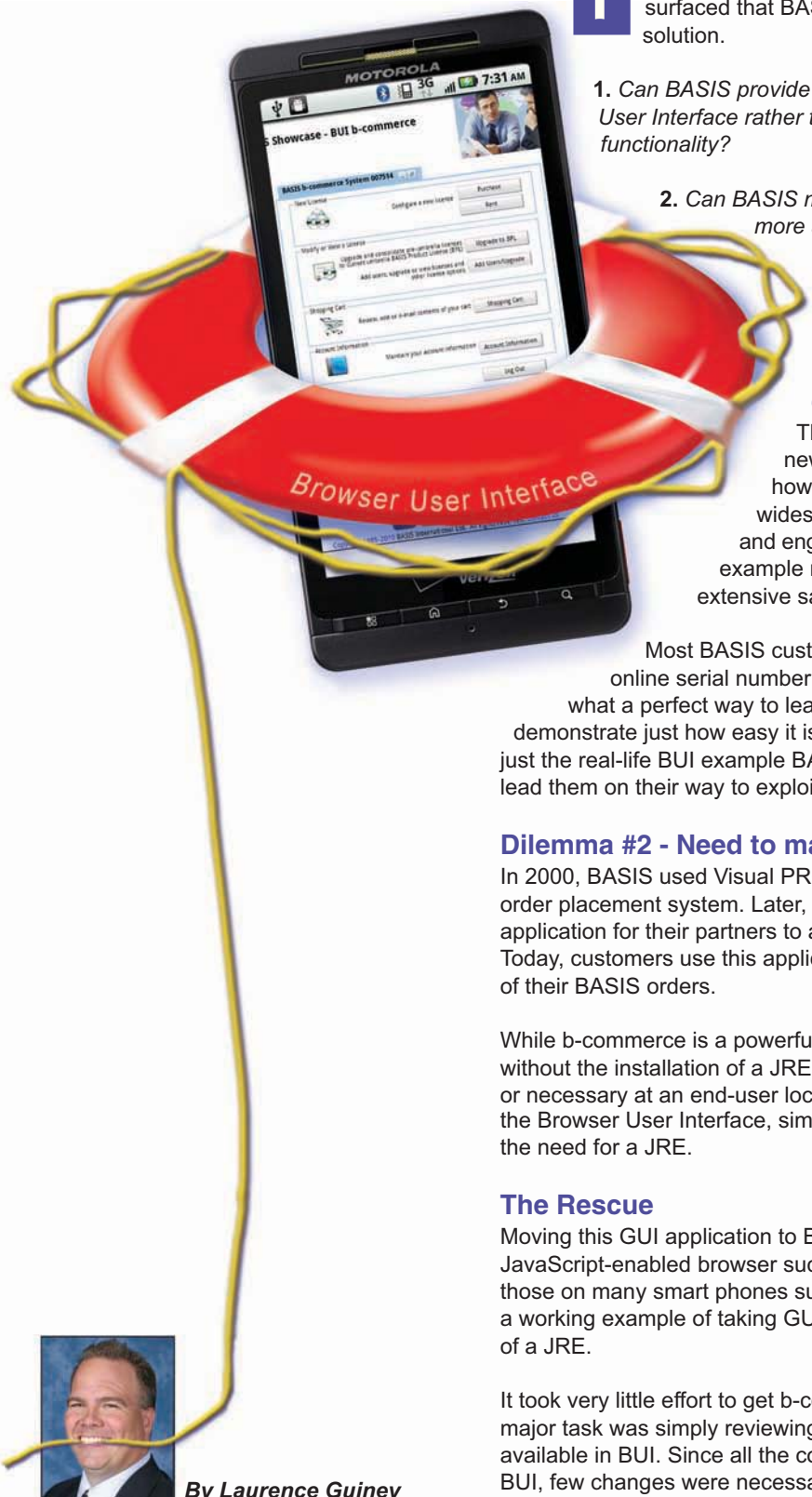
BUI to The Rescue

For over 25 years, BASIS has committed to listening and responding to the needs of their customers. Recently, two unrelated concerns surfaced that BASIS was able to address with one very exciting solution.

1. *Can BASIS provide a real-world working example of the new Browser User Interface rather than the typical demo application with limited functionality?*

2. *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 the new BUI (Browser User Interface) came to the rescue!



Dilemma #1 - Need 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; it deserved a more extensive sample to showcase its capabilities.

Most BASIS customers are already familiar with b-commerce, the online serial number information and order placement application, so what a perfect way to lead by example. 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.

Dilemma #2 - Need to make 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, www.poweredbybbj.com, to place the majority of 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 BASIS to run the application on any JavaScript-enabled browser such as Explorer, Firefox, Chrome, and Safari, including those on many smart phones such as the iPhone, Droid X, or HTC Desire. This was a working example of taking GUI to BUI while eliminating the need for the installation of a JRE.

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

By Laurence Guiney
Senior Account Manager

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 a customer base thrilled with the option to access it from any device without the need for an extensive JRE download. Now, customers can run it in their office or on the go, running a browser on either a PC or mobile device. 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 quickly and easily migrated a GUI application originally developed in Visual PRO/5 to a BUI application, resolving two very valid concerns. 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. ■

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

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:

```
j                                >
```

For example,

```
j                                              r
                                                   r
```

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:

```
                                  [-   ]                              >
 -r record
 -p playback
     ck          fil          gs,     ha    us    ca       cha  ge   he
```

For example,

```
j                                  r
j                                                                   s
j                                                                   s
j                                                                    s
```

>>

By Brian Hipple
Quality Assurance Supervisor

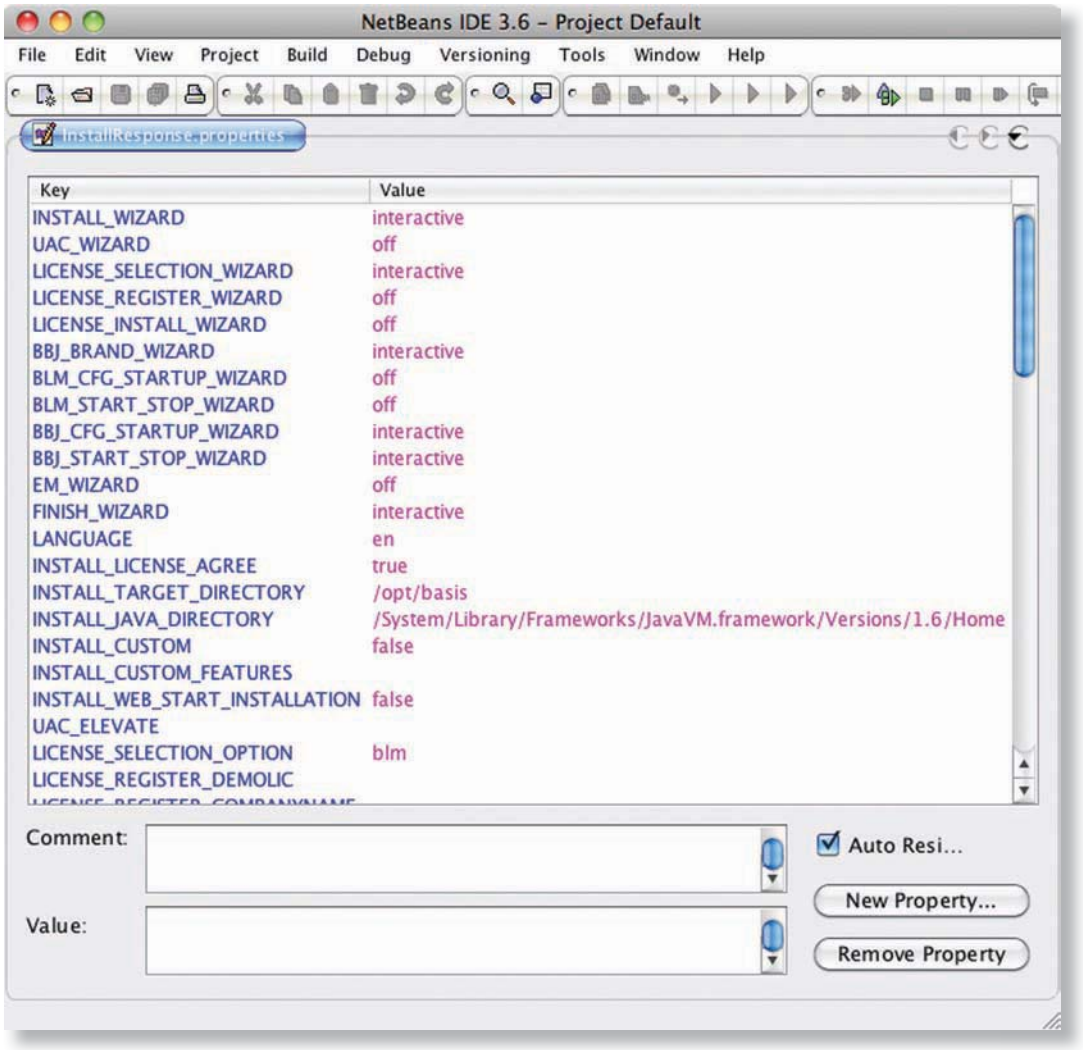


Figure 1. Example of a response file

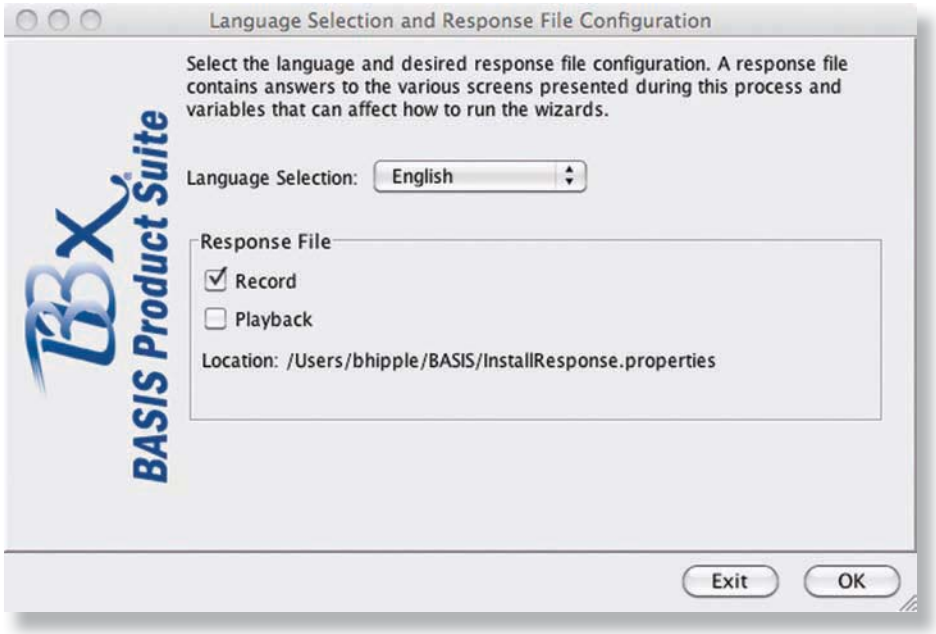


Figure 2. The installer's initial screen displaying response file options

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**.

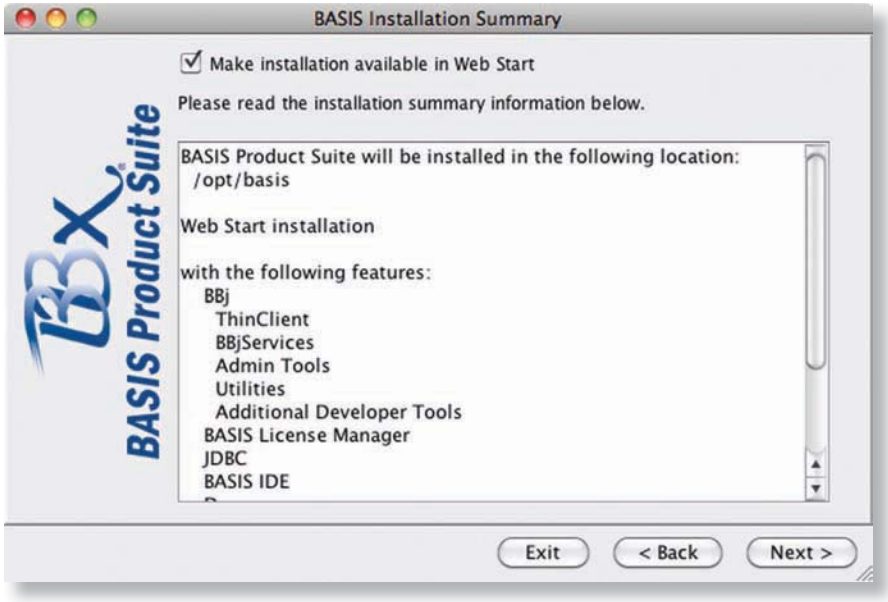


Figure 3. The summary screen displaying the Web Start Installation option

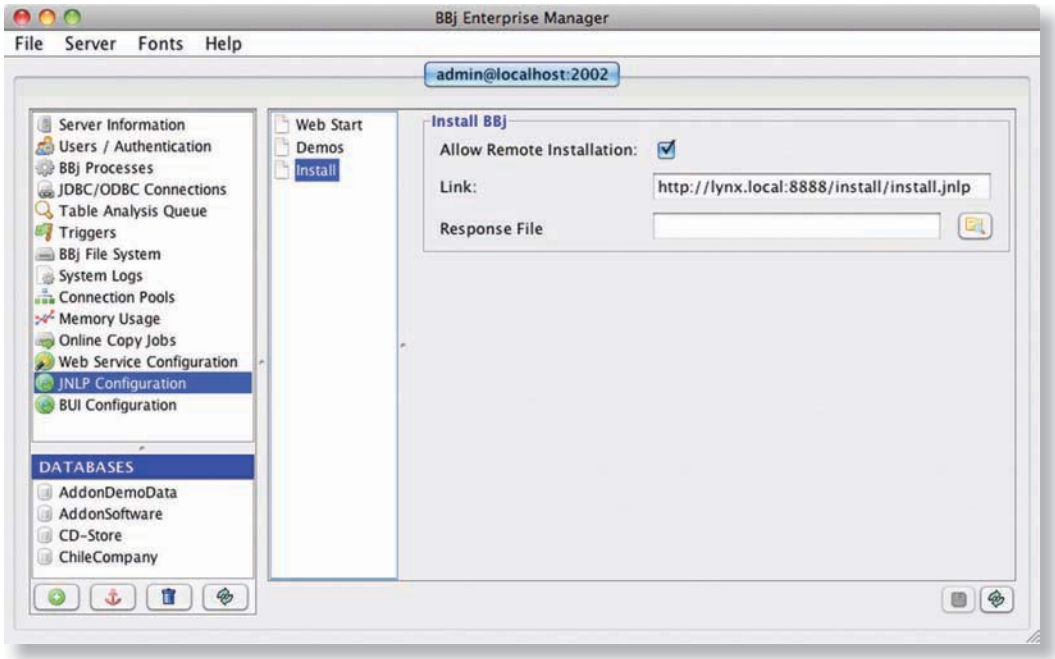


Figure 4. Enterprise Manager's Web Start Installation configuration

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. ■

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 Secure Sockets Layer 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). Built-in demos make useful applications available in many forms. BASIS also added many user interface improvements for configuring Web Start and Web Services.

Read on to discover some of these new elements experienced or "tasted" in today's nicely aged BBj.

Secure Sockets Layer - Subtler

Secure Sockets Layer (SSL) is a key part of protecting your user's information on the Internet. 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 one of the many providers 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.

By Jason Foutz
Software Programmer

EM and BUI - Bigger Bouquet

Another great 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 simply use EM to select your config file and program as well as other options. EM then automatically 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 on the Web using SSL protocol. Existing features and demos are easier than ever before to use and configure. Jetty enhances the ever improving 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 Progr

Partnership | Language/Interpreter | DBMS | Development Tools | System Administration | Building Blocks

BASIS 25th Anniversary — the next generation

- **TechCon/AddCon** — May 2011
- **BASIS products** — 
- **BASIS/EMS Diamond Work Group Conference** — October 2010
- **PRO/5 10.0, Visual PRO/5 10.0, AddonSoftware 10.0** — July 2010
- **BBj 9.0, Barista 9.0** — March 2009
- **AddonSoftware 9.0** — July 2009
- **TechCon/AddCon** — September 2009

- **AddCon** — November 2008
- **AddonSoftware® 8.0** — August 2008
- **PRO/5 8.0, Visual PRO/5 8.0** — June 2008
- **PRO/5 6.0, Visual PRO/5 6.0** — August 2006
- **Audev Germany partnership** — April 2007
- **BBj 6.0** — June 2006
- **TechCon** — May 2006
- **20th anniversary BASIS products** — October 2005

- **BBj 5.0** — May 2005
- **PRO/5 4.0, Visual PRO/5 4.0** — July 2002
- **BASIS products ship on CD** — March 2003

- **BBj IDE, BBj 2.0** — April 2002
- **TechCon** — May 2001
- **TechCon** — September 1999
- **PRO/5 3.0, PRO/5 Data Server 3.0, BASIS ODBC Driver 3.0** — October 1999
- **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

- **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
- **BBXchange Conference** — September 1993
- **BITS Conference** — April 1993
- **Coyote College Training** — March 1993
- **BITS Conference** — November 1992
- **BASIS launches Business Basic Extended BBX Level 1** — October 1985

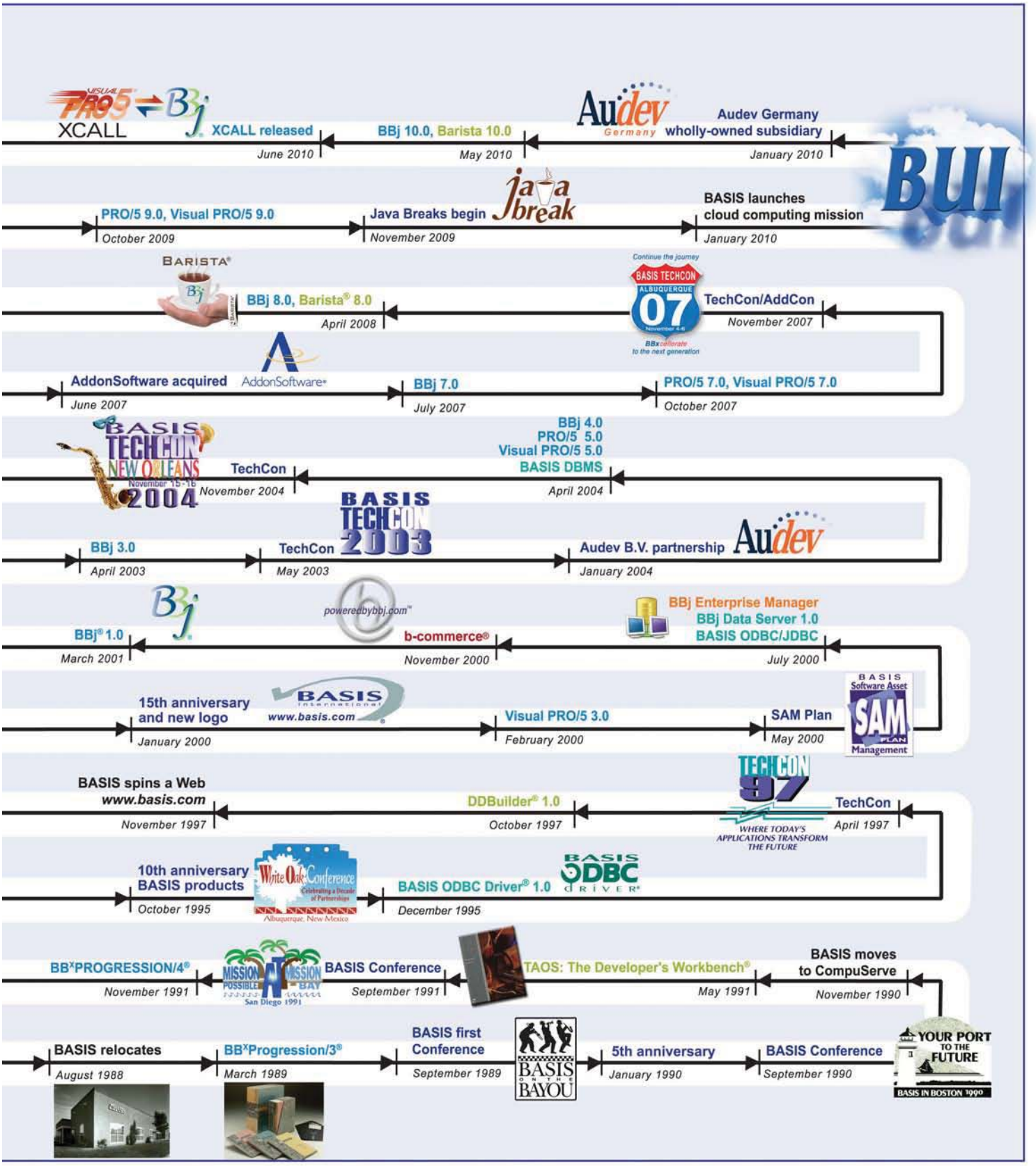
- **BASIS moves to electronic BBS** — December 1986
- **BBXProgression/2®** — May 1987
- **New name and logo** — January 1988
- **BASIS Incorporated founded** — January 1985

BASIS INTERNATIONAL

PRO5® VISUAL PRO5® BBj® BARISTA®

Return to TOC

ression... the BBx Generations



AddonSoftware®

BASIS International
www.basis.com®

Return to TOC

Preserving Your Customizations

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.

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. >>

By Chris Hawkins
Software Developer

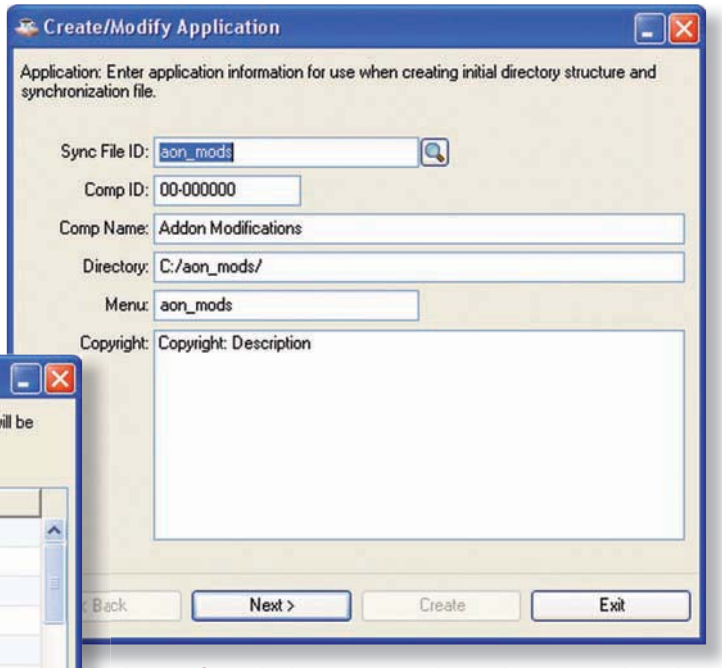


Figure 1. Create Application wizard collects information for the project

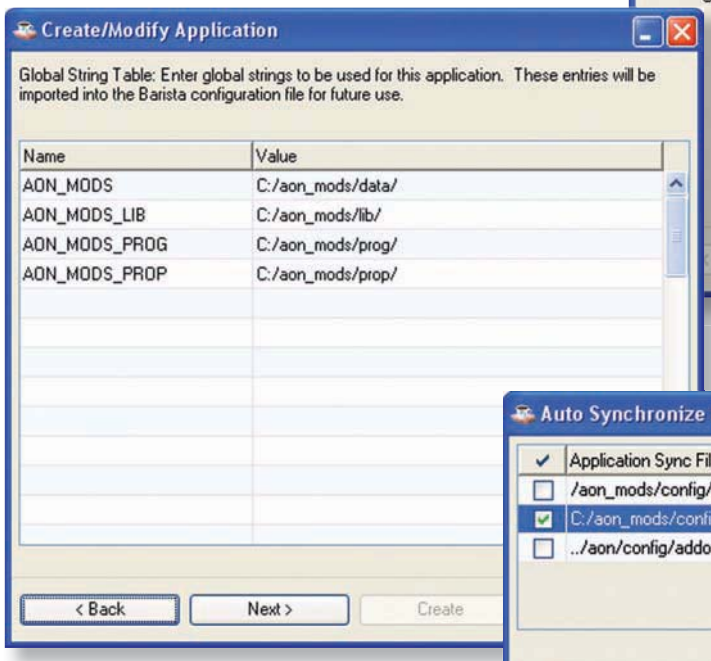


Figure 2. Barista provides default configuration settings

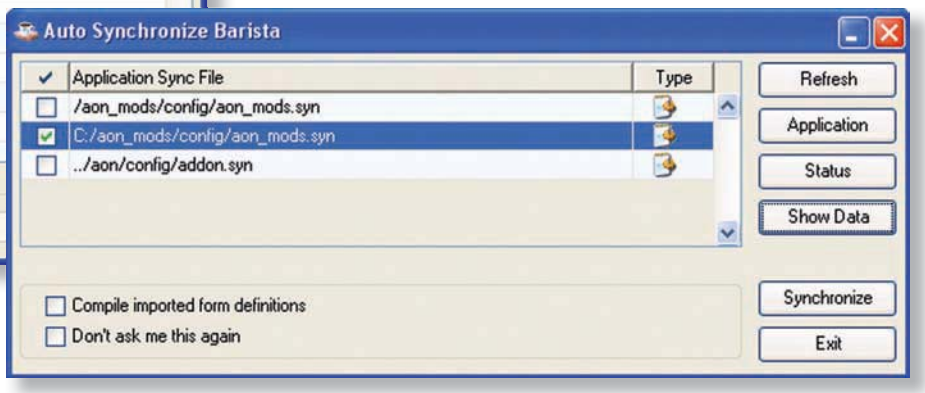


Figure 3. Auto-Synchronize new project into Barista

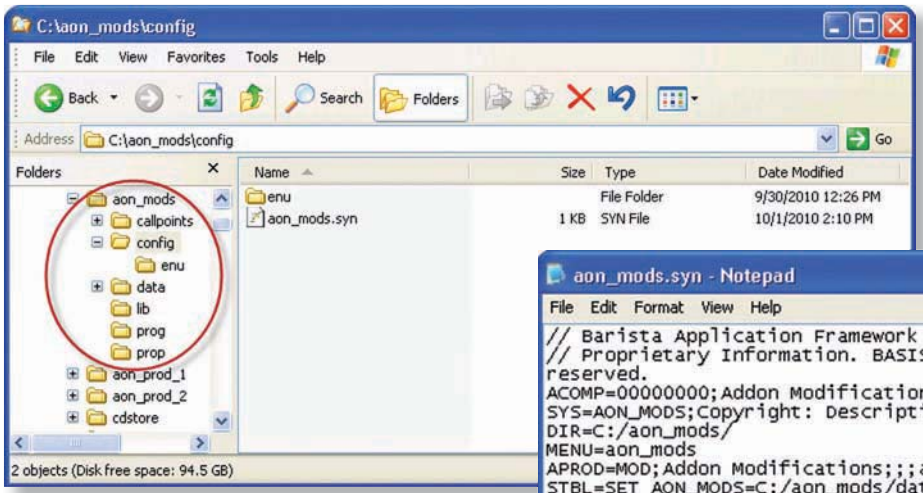


Figure 4a. Barista creates directory structure for new project

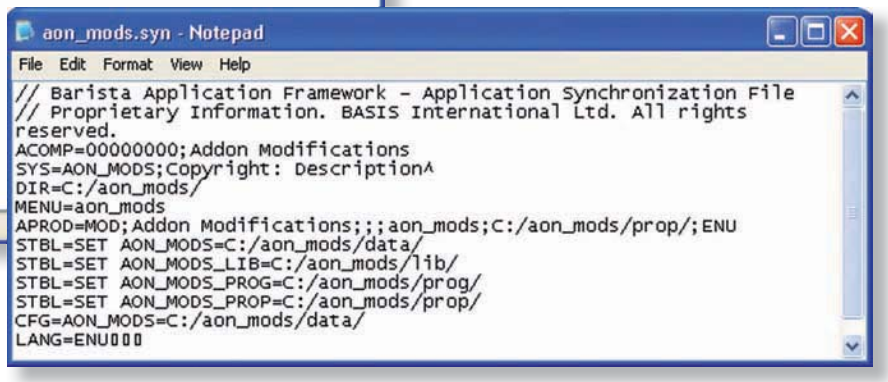


Figure 4b. Information supplied during wizard is captured in .syn file for new project

>>

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.

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**). >>

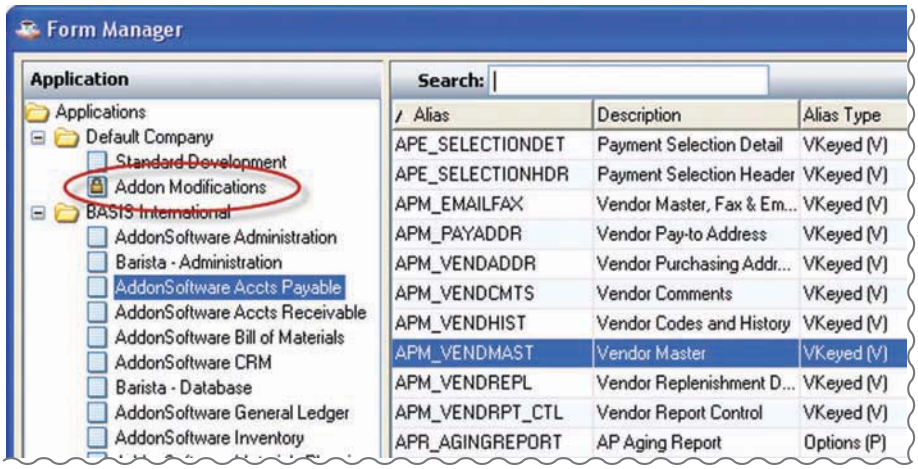


Figure 5. Development in replication mode

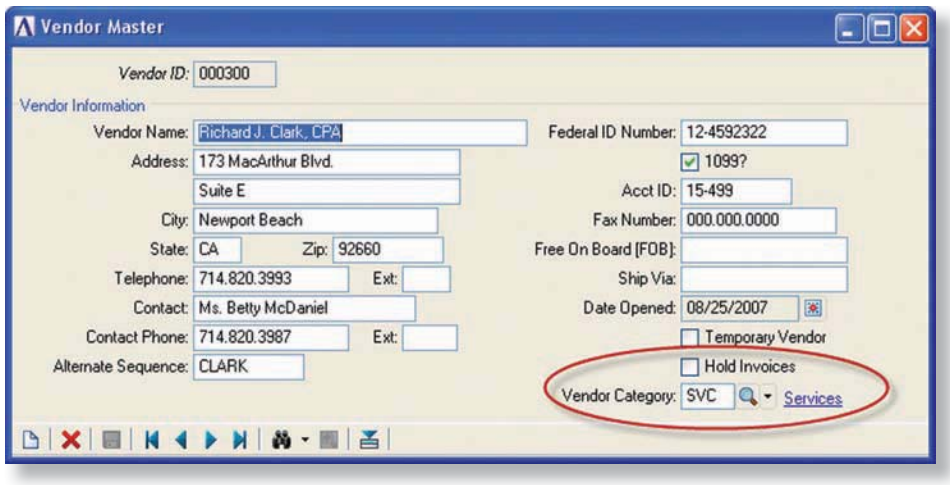


Figure 6. Revised Vendor Master form: moved Hold Invoices to the right and added new Vendor Category field

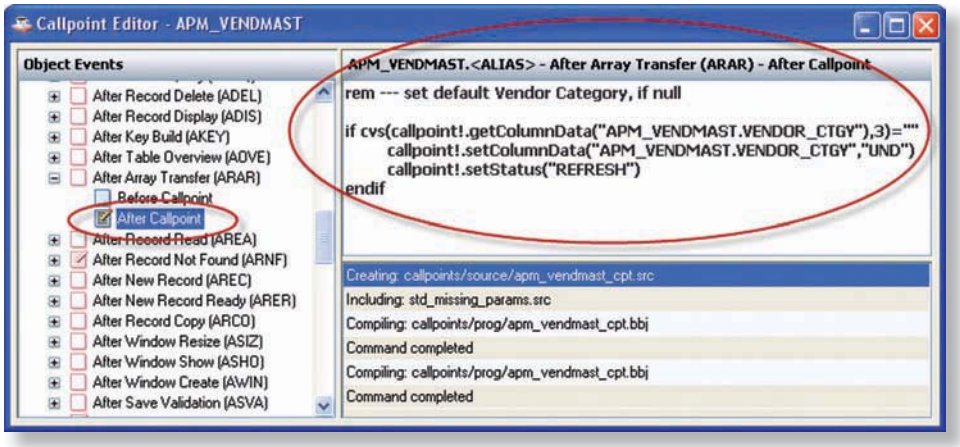


Figure 7. Add custom business logic using the Barista Callpoint Editor Before/After callpoints

To run your code only, that is, instead of the standard callpoint code, use the Before callpoint along with the method `callpoint!.setStatus("SKIP")`.

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**. 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 **>>**

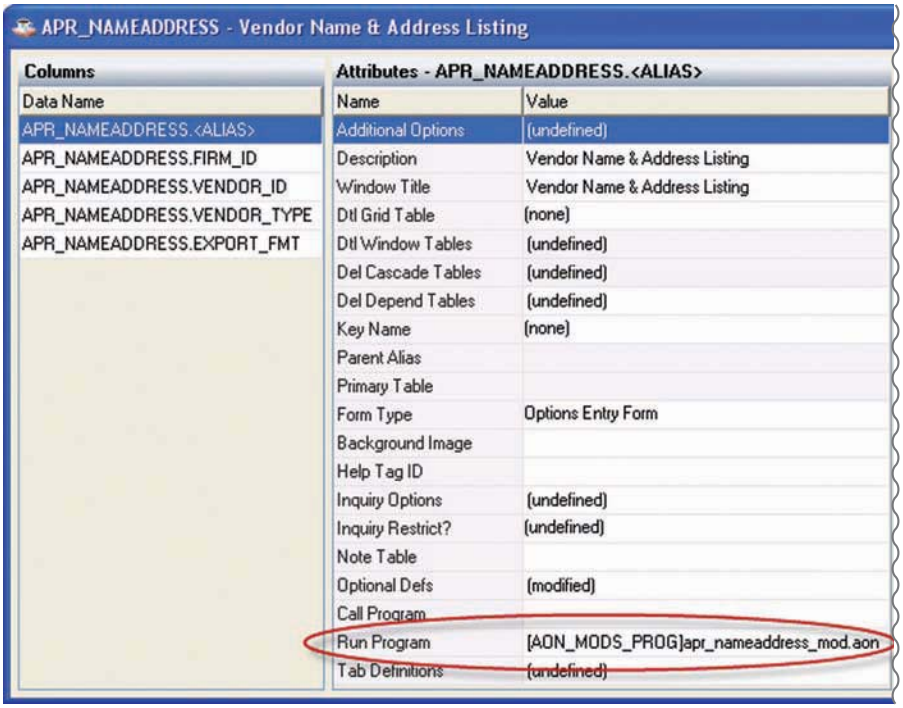


Figure 8. Changing the "Run Program" tells Barista to run our customized report

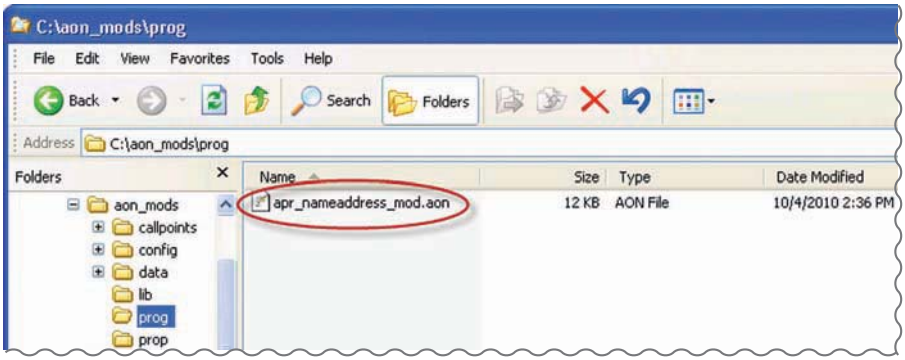


Figure 8a. Place back-end code in your project's "prog" directory

Figure 9. Customized Vendor Name and Address Listing

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.

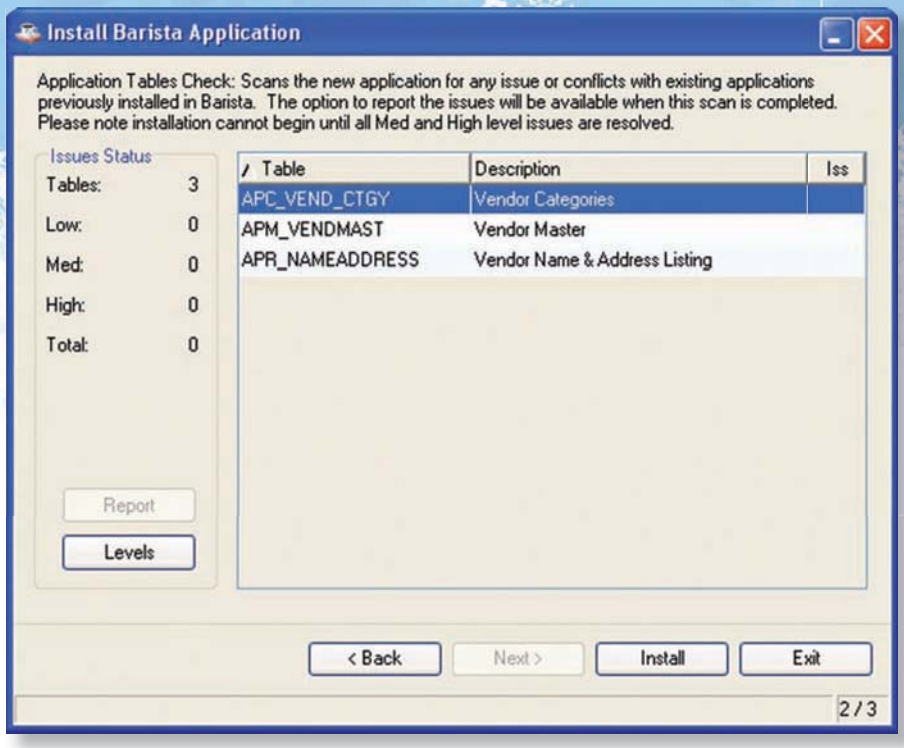


Figure 10. Barista Install Application Wizard

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! ■

BBj PDFs - Perfectly Displayed Formats

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

With the cross-platform success of PDF files, this is a great format for creating and distributing reports for your Windows, Mac, and Linux users alike. In the past, it was possible to use operating system specific print drivers to create a PDF file but it was often challenging and perhaps expensive and required an extra step during the installation process. BASIS built PDF support right into BBj so there are no extra steps to create PDF documents and reports. This article takes a closer look at generating these PDF files in all three ways.

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! ■

By Jason Foutz
Software Programmer

For more information on
- JasperReports, visit jasperforge.org/projects/jasperreports
- Report writing, read the Advantage article *Recipes for Successful Report Writing* links.basis.com/09reportwriting
- BBJasperReport::exportToPDF, refer to the online documentation links.basis.com/2309

AddonSoftware's Barista

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.

By Paul Yeomans
Vertical Market Account Manager

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 28).

AddonSoftware is the Answer

Neither overly complex nor costly, and able to preserve your customizations through >>

Building Blocks Strategy

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

> "margins as high as 100%"

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.

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.

By Jeff Ash
Software Engineer

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.

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 >>

```
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

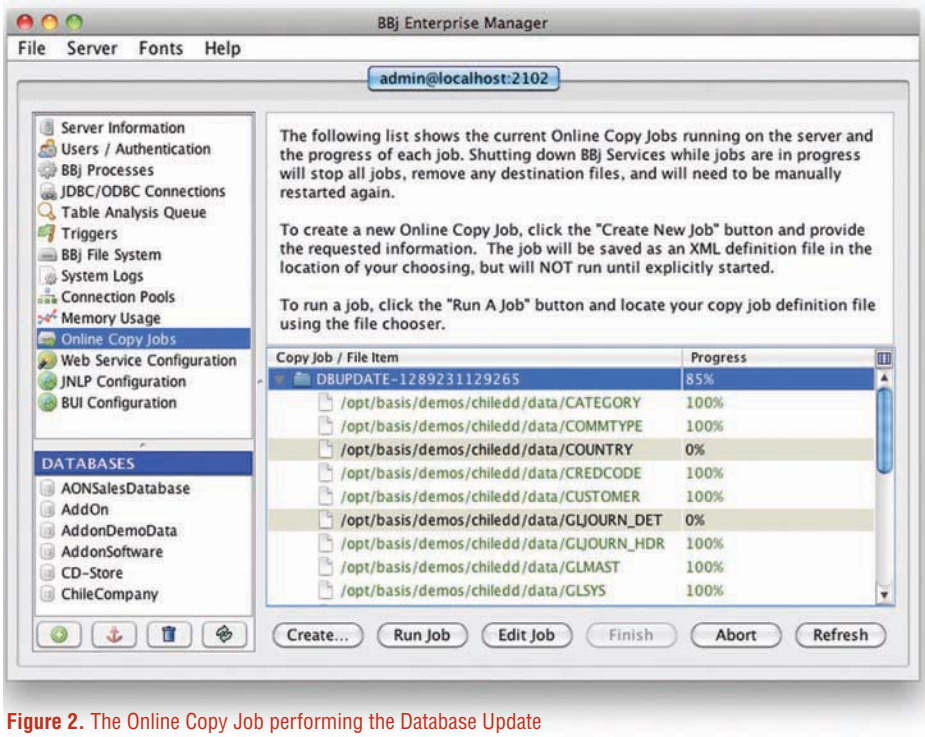


Figure 2. The Online Copy Job performing the Database Update

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!

Not only can an administrator use the handy wizard found in the Enterprise Manager, but 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. This flexibility makes this new feature very powerful for administrators and developers alike. ■

- Read the online documentation for Admin API Javadocs at links.basis.com/iocgg
- Try the sample at links.basis.com/10dbcode

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.

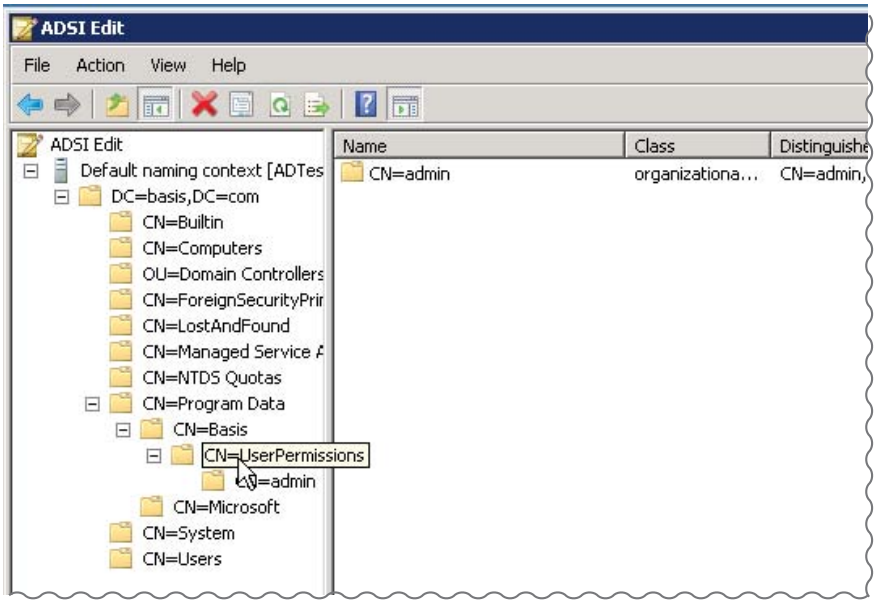


Figure 1. Create a sub directory structure with class type "container"

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.**

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. >>

By Jeff Ash
Software Engineer

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.

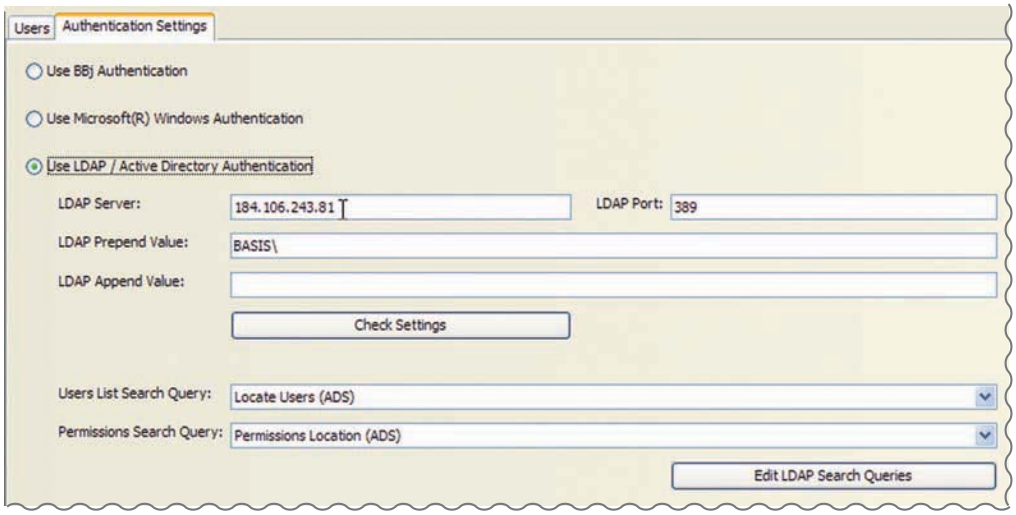


Figure 2. Active Directory Authentication configuration in the Enterprise Manager

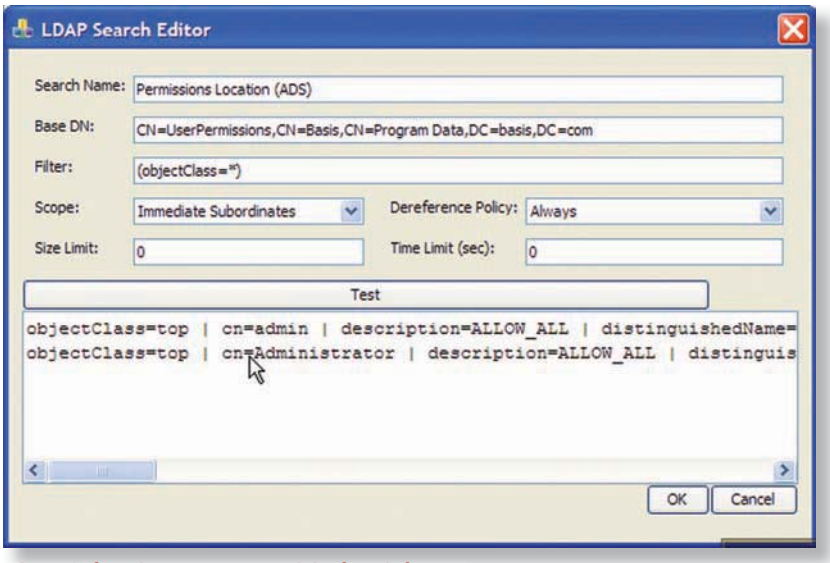


Figure 3. Sample text to enter in LDAP Search Queries

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
```

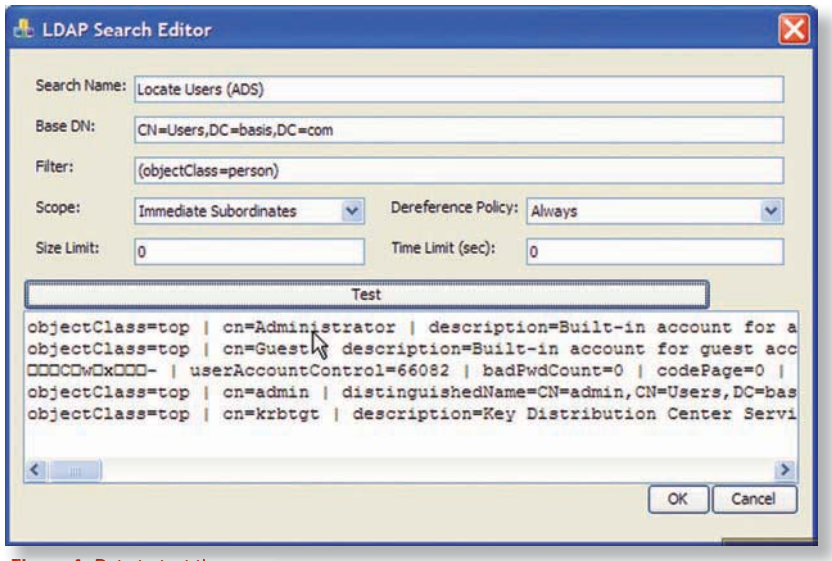


Figure 4. Data to test the query

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.

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. >>

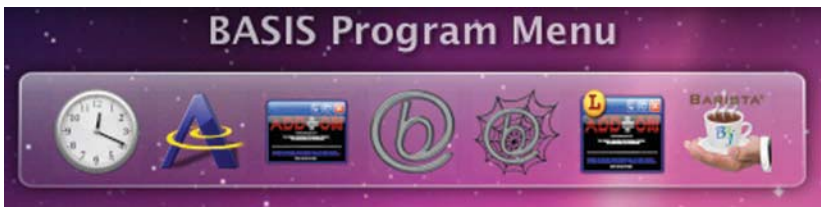
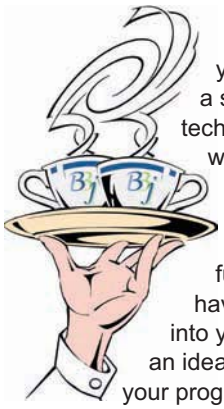


Figure 2. The BASIS Program Menu LaunchDock

By Brian Hipple
Quality Assurance Supervisor

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/ebjgg
- Learn about these integrated BASIS solutions
  - Forms –
    - *BBj Form Printing* links.basis.com/03bbjforms
    - *Dishing up Forms* links.basis.com/08forms
  - Forms/PDF – PDF *Now Also Means Perfectly Displayed Forms* links.basis.com/05pdf
  - PDF – *BBj PDFs - Perfectly Displayed Formats* links.basis.com/10bbjpdf
  - Fax and e-mail – run the demos under "Demo LaunchDock" included in the BBj download

NEWS

Mining the Future with Diamond 725

BASIS International Ltd. and EMS Healthcare Informatics co-hosted the Diamond Work Group Conference in Albuquerque immediately following the Albuquerque International Balloon Fiesta. The timing of this conference in early October afforded some of the attendees the opportunity to arrive a day or two early to enjoy this world famous event. This year's Fiesta was one of the best, with great weather and over 500 balloons for spectators to enjoy and photograph, some with amazing shapes.

By Gale Robledo
Account Manager

The theme for this conference was "Planning for the Future with Diamond® 725" with the goal to work, learn, collaborate, and network as a group. And work, learn, collaborate we did, with a productive two-day exchange that resulted in much collaborative work for BASIS and EMS to complete in the coming months. Together, we addressed application changes necessary for today's healthcare requirements, enhancements that BASIS technology provides, and a top-to-bottom support model that provides Diamond users with a secure future.

Keeping up with the healthcare industry regulatory changes, security and data exchange requirements, and technology choices is a very tall order for customers running on BASIS technology. Attendees discovered that BBj provides the necessary tools and features to get the job done, today and far into the future. User groups like the Diamond Work Group Conference provide the opportunity for peers and technology providers to come together and help each other stay abreast of the ever-changing industry.

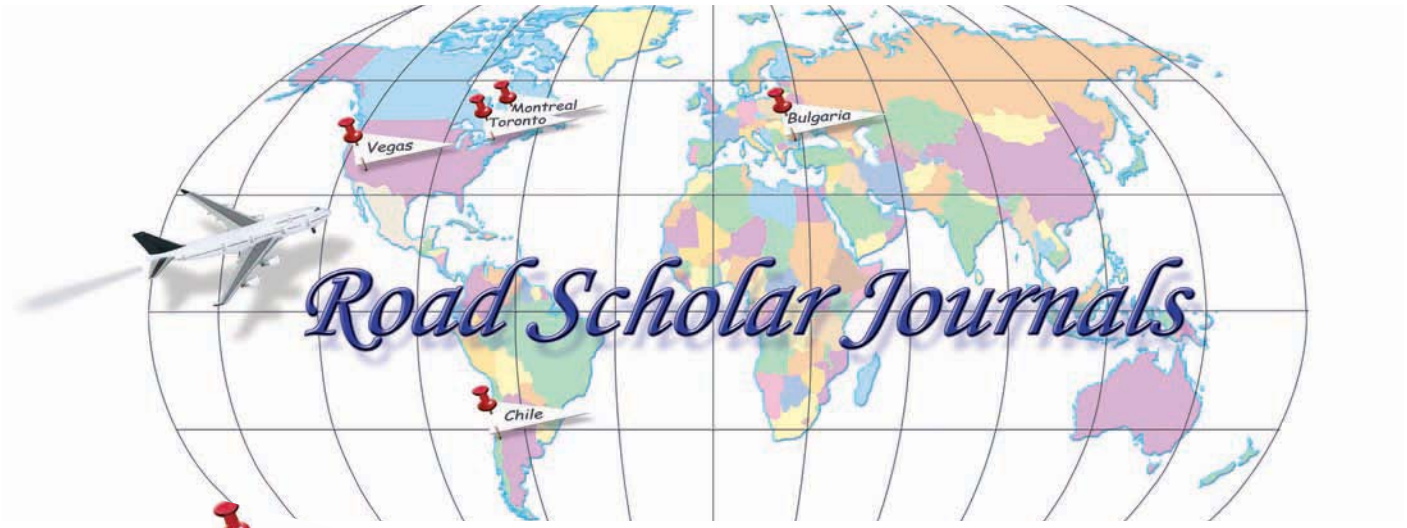
BASIS welcomes the opportunity to partner with resellers and users, and to participate in strategy sessions and technology reviews. Such a partnership helps to grow and nurture successful businesses running on BASIS technology. BASIS also provides professional services to jump start your BBj and Barista projects. The BASIS business development team stands ready to strategize with and assist any of our partners with their Business opportunities. Call BASIS Sales for assistance in any of these areas. ■

Visit BASIS Sales at www.basis.com

EMS Healthcare Informatics

BASIS International

Planning for the Future – Diamond Work Group Conference



Bulgaria

BASIS Successfully Prospects in Bulgaria

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! ■

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.

By Patrick Schnur
Marketing
Public Relations

Barista Gets a Hot Chile Reception in Santiago

This year has been full of significant events for Chile. In February, a major earthquake struck the country. In June, Chile's soccer team won two games at the World Cup, their first tournament victories in 48 years. In August, the collapse of the mine near Copiapó left 33 miners trapped, resulting in a rescue that kept the world in suspense for 69 days and showed off the ingenuity of Chilean engineers.

In addition, 2010 saw the Chilean economy return to strong growth with an estimated GDP growth of over 5%. This is quite an achievement considering the recession in the rest of the world and makes Chile the strongest economy in Latin America. This growth extends to the information systems industry as well, with significant innovation seen in the development of computer software and applications.

BASIS and ESS are no exceptions to 2010 achievements as they jointly hosted a very successful BBj/Barista® 10.0 Road Show at the Four Points Sheraton in Santiago. The event began with a half-day product presentation by Nico Spence, BASIS CEO, who highlighted the BASIS product strategies and demonstrated the new BBj 10.0 functionality and development with Barista. Nico also presented practical alternatives to migrate existing PRO/5® and Visual PRO/5® to BBj®. Jesús Alvarez followed with an overview of Barista, explaining its evolution and emphasizing how it allows developers to take advantage of modern language features while significantly reducing the time to develop business applications. Next, William Hernández demonstrated how to develop a simple application with Barista. Attendees were pleasantly surprised by Barista's functionality and productivity.

Photo credits: José Aguilar

A full room awaits the presentations from Nico Spence, Chairman & CEO, BASIS; and Jesús Alvarez, President, ESS and William Hernandez, Senior Consultant, ESS.

Nico discusses current strategies and the future of BASIS.

The BBj training led by William Hernandez was well received.

Two days of training in Barista development, taught by William Hernández, followed the Road Show. By covering the fundamentals of application development with Barista, William helped the developers gain the skills required to start new applications and migrate existing ones to this new development platform. Attendees particularly enjoyed the data dictionary centric design, parameterization capabilities, security system, and extensibility via call points. Barista allows developers to concentrate most of their development time on implementing business rules rather than managing the user interface. In fact, several trainees left committed to start new applications using Barista, BASIS' hot application development tool.

Special thanks to SSL Computación for their assistance in marketing the Road Show and for their sales and support efforts with BASIS products in Chile. ■

By Jesús Alvarez
ESS, BASIS
Product Distributor

Chile was one of the first countries outside the US to use BASIS programming languages for development. Continuing a 25 year run, they are actively developing with BASIS tools in a country that is an important cornerstone for BASIS in the South American market. Several financial firms use BBj to handle investment activities, including trading systems for the Chilean stock market. Many other vertical applications are written with BASIS languages for supermarkets, pharmacies, customs brokerages and other industries.

Canada's Software Showcase 2010

Face-to-face interaction is a highly valued commodity within the Canadian software industry. In keeping with this value, Descore, Inc. hosted a pair of events this September that brought together Canadian resellers and U.S. software developer communities. Named Software Showcase 2010, this event created a unique opportunity for resellers to network with one another and help close the gap between the developer and the reseller. Attendees had the opportunity to learn first-hand about the newest functionalities added to their chosen developer's product lines and to speak directly with representatives of both BASIS International Ltd. and Synergetic Data Systems.

By Dave Foster
President, Descore Inc.
BASIS Product Distributor

First Stop, Montreal

On September 27th, the Marriott Courtyard located near the Montreal airport was the venue for this well attended event, despite construction on many of the neighbouring streets. Nico Spence, Chairman and CEO of BASIS, came well prepared to demonstrate the newest BBj® features. He introduced BUI (Browser User Interface) and demonstrated its functionality, then re-introduced Barista to our customers. The foreign language capabilities of Barista was of particular interest to our French-Canadian VAR community who wanted to move their already multilingual applications forward to a graphical presentation in the fastest possible way. Barista is just what the doctor ordered!

Final Stop, Toronto

Descore hosted the last showcase on September 29 at the Thornhill Golf and Country Club in north central Toronto. The ambiance of the clubhouse continues to create an air of participation and networking that has proven, year after year, to be a well received and desirable venue for our Toronto-based customers.

We asked our customers to provide feedback to help us determine future needs. Many requested more technical focus and less marketing-oriented sessions with a chance for dynamic conversation about BASIS technology and their individual applications. Nico included one of the shorter Java Break sessions in his presentation to demonstrate the step-by-step process of moving a character-based application to a graphical application with Barista and the ease with which new BBj features integrate into existing applications using Barista. This simple presentation displayed the value of the Java Break sessions (go to links.basis.com/JavaBreak). We expect greater participation from Canadian VAR's in future sessions.

The Results

Through these showcase events and individual training sessions, Descore succeeded in moving good applications forward technologically, making them better, using object-oriented functionality in BBj …the Java-based object-oriented extension of Business BASIC. ■

OSAS and BASIS in the Bright Lights of Las Vegas

Once again, Open Systems, Inc. continued their reputation for excellence with the production of their annual conferences. They put on quite a show this September...for a full week in Las Vegas! Act 1 was the Partners in Profit conference for the reseller channel, followed by Act 2 - Customer Excellence conference - for the end user channel. This format of back-to-back conferences maximized the networking opportunities and the ability to educate all of their customers.

It was great to visit with Open Systems resellers, meet end users, and discover what is on the horizon for the OSAS product. Dave Link, Vice President of OSAS Product Development, shared some exciting new directions for his development plans and their use of some of the latest BBj features. We are thrilled that the OSAS product has closely followed our releases of BBj, allowing them to implement our new features as quickly as possible and we were especially pleased with the announcement that OSAS 8.0 will be a BBj-only release.

By Gale Robledo
Account Manager

BASIS was privileged to be an exhibitor and presenter at both the Partners in Profit and Customer Excellence conferences. BASIS' dynamic duo - Nico Spence, CEO, and Dr. Kevin King, President - did an outstanding job presenting our experience of moving our entire enterprise to Cloud Computing; using BBj Web Server and iReports, and many other features that enhance the OSAS experience.

Dr. Kevin King and Nico Spence present Cloud Computing

A conference highlight each year is the Top 25 awards banquet when Open Systems recognizes their top resellers for their hard work and success. Response Computer Group (RCG) from Milford, Delaware, earned this year's top honor. It was a return trip to the podium for RCG with an encore performance. RCG's product sales come mainly from the OSAS product line built on BASIS technology. Read more about their organization in the *BASIS International Advantage* feature article "RCG Uses Marketing Skills to Achieve OSAS 'Top Dog' Status" (see links.basis.com/07rcg).

Congratulations to Response Computer Group on another successful year using OSAS and BASIS products to enrich and grow their customers' businesses. And a special "Thank you" to Open Systems for hosting such a great conference. ■

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.

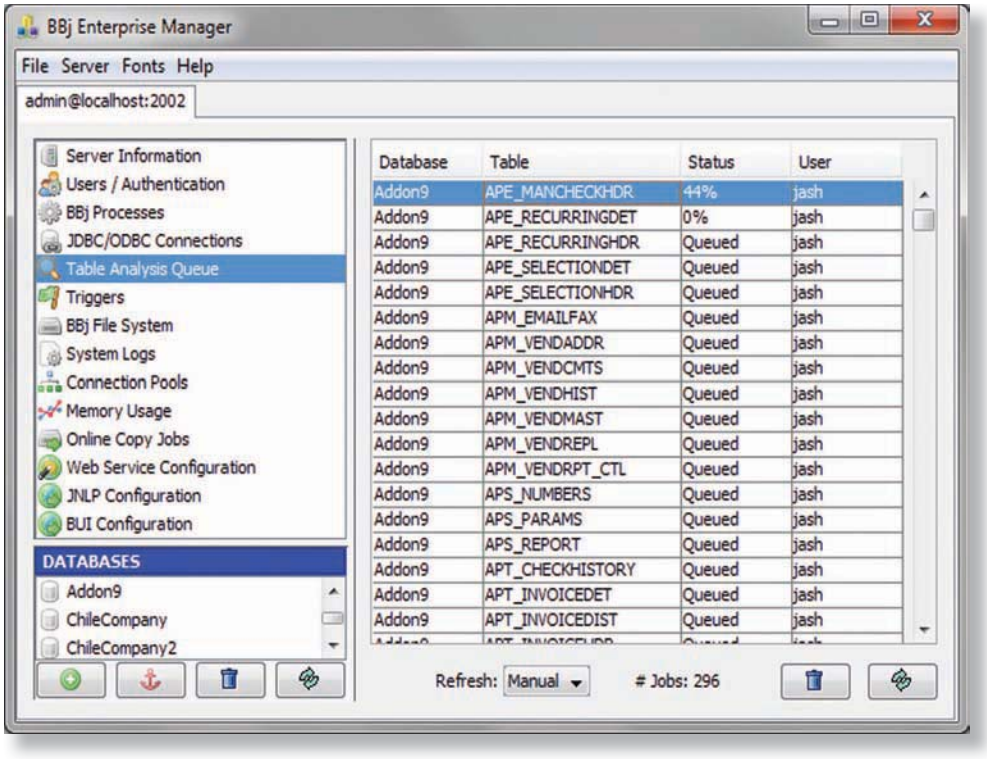


Figure 1. The Table Analysis Queue in Enterprise Manager

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

By Jeff Ash
Software Engineer

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. ■

Enterprise Manager Functionality Free at Last

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. >>

```
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

By Jeff Ash
Software Engineer

```
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 code that demonstrates how to create an online copy job programmatically

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.

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/10emcode

Photo credit: Las Vegas News Bureau

Tech Resource Zone

Virtual Licensing
Key to Apps in the Cloud

It's no secret that many organizations are going virtual. Whether they are using VMWare or Xen to run test Linux servers under their desk or a production system in the cloud, 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 BASIS 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 hostid 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 - one that is not susceptible to a changing hardware profile.

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. >>

By Bruce Gardner
Technical Support Supervisor

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

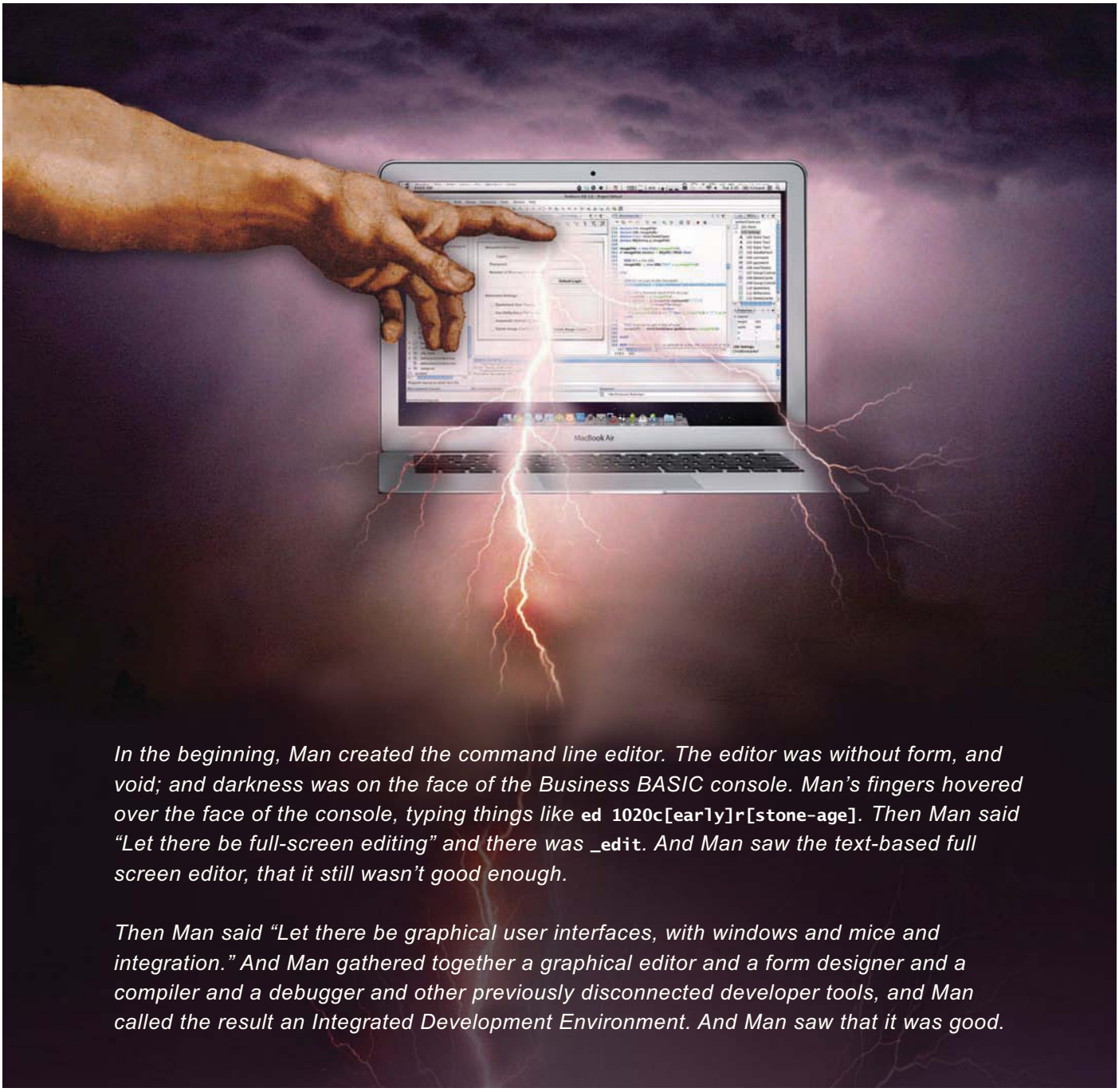
Q. ***How do I install and use a virtual license?***

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.

Summary

With virtual licensing, BASIS has removed the last obstacle to taking your complete operation to the cloud. Spend just a little time experimenting with VMWare, Xen, or cloud infrastructures such as Amazon's EC2 and be convinced of the benefits of virtualization. Then exploit this valuable resource for your cost-effective disaster recovery, load balancing, and reliable service. The sky is the limit! ■

- For more information, read the online BASIS documentation links.basis.com/001
- 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



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.

BBj IDE in the Beginning - Then and Now

Man is not yet finished with creation, of course. At BASIS we are constantly engaged in making improvements to our own Integrated Development Environment (IDE). 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.

By Mike Phelps
Software Programmer

Read on for the IDE's most helpful features, what it takes to get it running, and 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 modules that 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. For better or worse, modern IDEs are large, complex *desktop* applications dedicated to individual users who each have their own relatively powerful machine. We have never attempted to use the IDE with a multi-user configuration, or received feedback from anyone who has, but our collected wisdom on the subject appears in the *Knowledge Base Article #01149*, called Multi-user installation of the BASIS IDE.

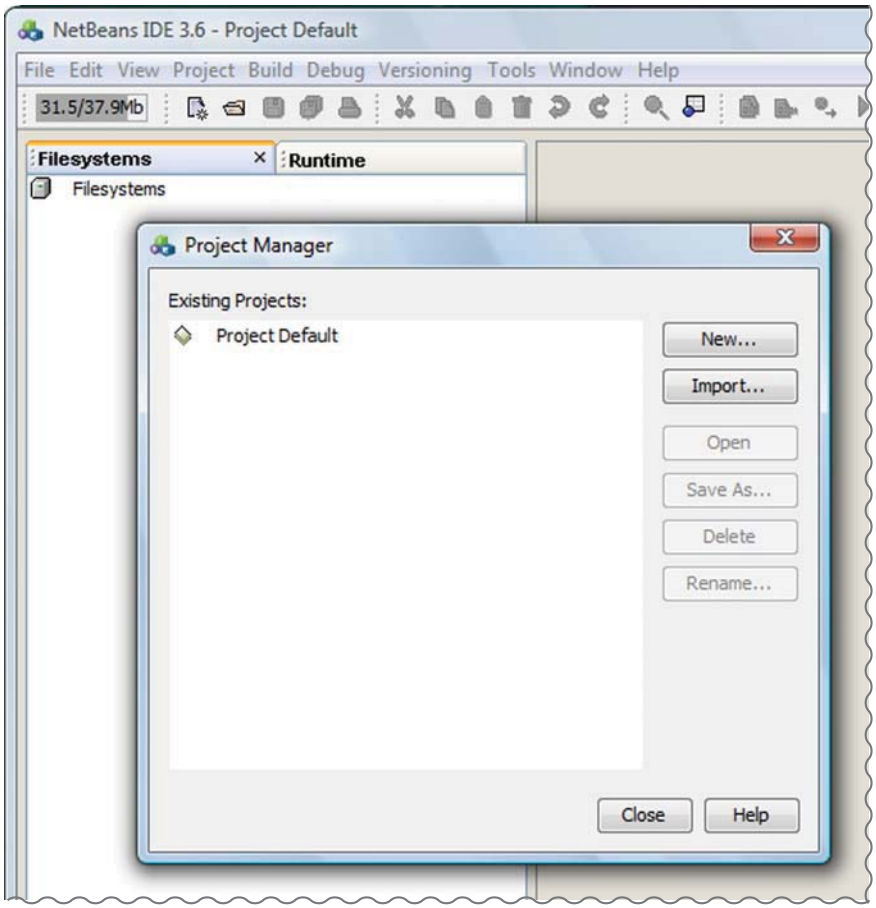


Figure 1. The "Create New Project" dialog

Another question we've been asked is "Can I download the NetBeans IDE from *netbeans.org* on the Web and then simply add BBj to it to get a working BASIS IDE?" Emphatically, this answer is no. The BASIS IDE uses an earlier version of NetBeans, which we've heavily customized with our plug-in modules, and which we maintain independently from the versions available from the NetBeans Web site. However, if you are already a NetBeans user with the latest version of the NetBeans IDE installed on your system, you can certainly install and use the BASIS IDE as well.

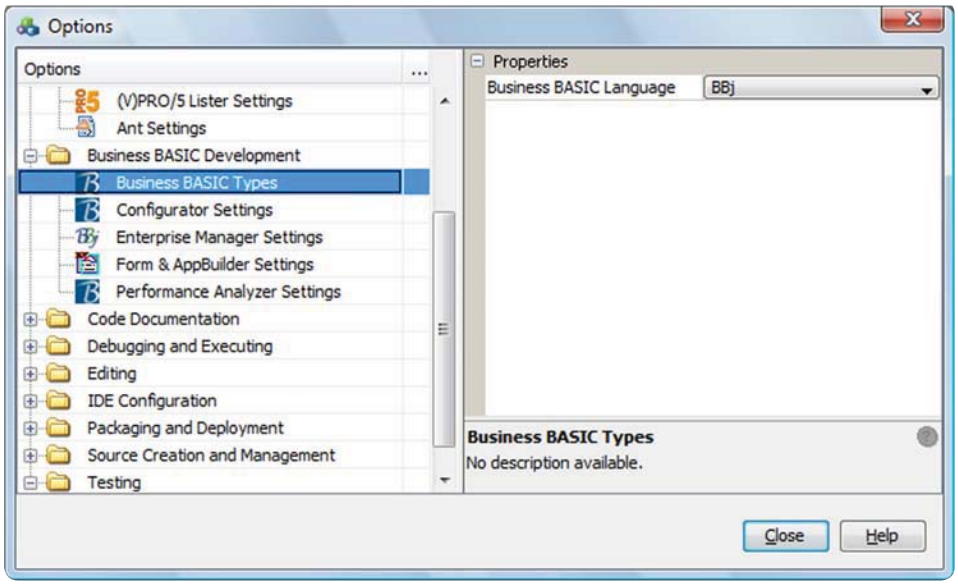


Figure 2. The Options window with Business BASIC Development expanded and Business BASIC Type selected

After successfully installing BBj and the IDE, what's next? The first major step is to define a Project that is a grouping of all the files and resources that belong to the application you are developing. Select **Project > Project Manager** from the main menu and press the [New…] button to create the new Project as shown in **Figure 1.** Every Project needs a unique name to distinguish it from all the others. (When you first start the IDE, you are placed in the *Default Project* automatically, but it is probably not the best idea to remain there.)

After entering a name, the IDE reconfigures to show a new empty project.

You need to specify what kind of BBx® project your application represents. Open the Options window via the Tools menu, then expand Business BASIC Development (see **Figure 2**).

The Business BASIC Language can be either BBj or (V)PRO/5 (Visual PRO/5® and PRO/5®). This setting controls which compiler, executor, and lister will be available to your project, and which set of icons will represent your files in the Explorer > >

Filesystems tab. Because the IDE cannot distinguish which program source files apply to (V)PRO/5 and which are meant for BBj based upon the files names or extensions alone, we recommend segregating BBj and (V)PRO/5 development into separate projects. If you mix the two different types of source in a single project, you must remember to change the setting of the Business BASIC Language property to the correct language type before compiling or running your code. The source files in the Explorer will all display the same icon to the left of their names (either all BBj or all PRO/5 icons), which may be confusing.

Next, start putting files into the new, empty project by associating directories with it. This is called "mounting filesystems." Right-click on the Filesystems node in the Explorer and select Mount as shown in **Figure 3**.

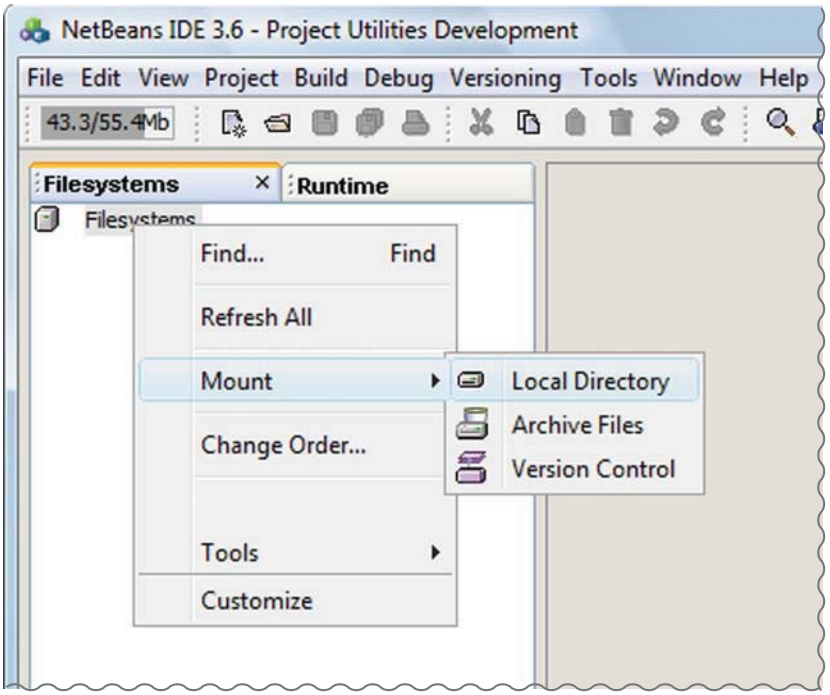


Figure 3. The Filesystems Mount submenu

The choices are *Local Directory*, *Archive Files*, and *Version Control*. A local directory is any directory containing your application's files that resides on a hard disk available to your computer. Archive files are Java .jar archives that your application may need. Version control refers to any local directories containing application files that are synchronized with a source code management (SCM) or version control system (VCS). The IDE can serve as an easy-to-use graphical interface for the CVS or Subversion SCM systems, sparing you the need to use a command line for the most common operations.

For the sake of performance, mount directories (see **Figure 4**) that contain only the source files and resources necessary for your application, and not directories full of other unrelated material (such as the root directory of the entire hard disk partition). The IDE examines each file in a mounted directory in order to assign a file type. If you mount drive C: or '/' as a single filesystem in the Explorer, this process will require a *lot of memory* and a *very long time* to finish.

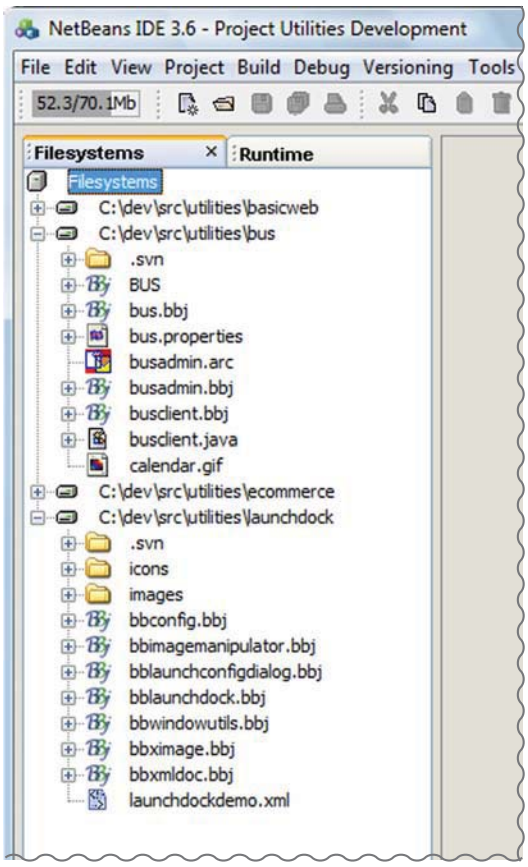


Figure 4. Mounted filesystems in the Explorer

If you are loading your new Project with an existing set of application files, there may be some additional configuration to do before the IDE will recognize your files as Business BASIC source files. The IDE distinguishes file types (and therefore what operations can be performed on those files) based on their file extensions. It is easy to tell if the IDE doesn't know what to do with your files: 1) You will see the 'empty page' icon just to the left of the file name nodes in the Explorer, and 2) when you right-click on a file name to open the popup menu, the first choice will be 'Treat as Text'.

Don't panic, this simply means the IDE doesn't have your file extension in its default list. You can fix this by going back to the Options window >>

as shown in **Figure 5**. Go to I**DE Configuration > System > Object Types > BBj Files**, then select the Extensions and MIME Types property as shown in **Figure 5**. Click on the ellipsis '…' box to open a property editor listing all the file extensions (including no extension at all) that are recognized as Business BASIC file extensions, and then add the extensions used by your files.

After adding the file extensions and closing the dialog and Options window, the Explorer will repaint itself. Your files will be assigned BBj or PRO/5 icons, depending on the Business BASIC type you have selected, and the popup menu will show a new set of choices.

You can create as many unique projects as you have applications to develop. There is no artificial limit. You may find that you need the same directory of files to be a part of more than one project at the same time, which is perfectly fine. We mentioned earlier that a best practice is to avoid mixing (V)PRO/5 and BBj development in the same project.

What if your BBj and PRO/5 code is stored side by side in the same directory or directory tree, and must remain that way? Just create two separate projects, set one of them as the (V)PRO/5 and the other as the BBj language type, then mount the same directories in both of them.

If you are just beginning an application development project, rather than loading an existing project into the IDE, you may just be mounting empty directories. In that case, how do you go about creating new files in the IDE?

Begin by selecting the mounted directory in the Explorer where your new file should be placed. Next select **File > New…** from the main menu, or right-click on the directory and select **New…** from the popup menu to open a wizard that guides you through the creation of a new file. The wizard offers templates (see **Figure 6**) for different file types with which the IDE can work.

After selecting one of the templates, you'll be asked for a name. Closing the wizard by pressing the [Finish] button creates the new empty file and automatically opens it in the particular IDE tool with which it is associated. **> >**

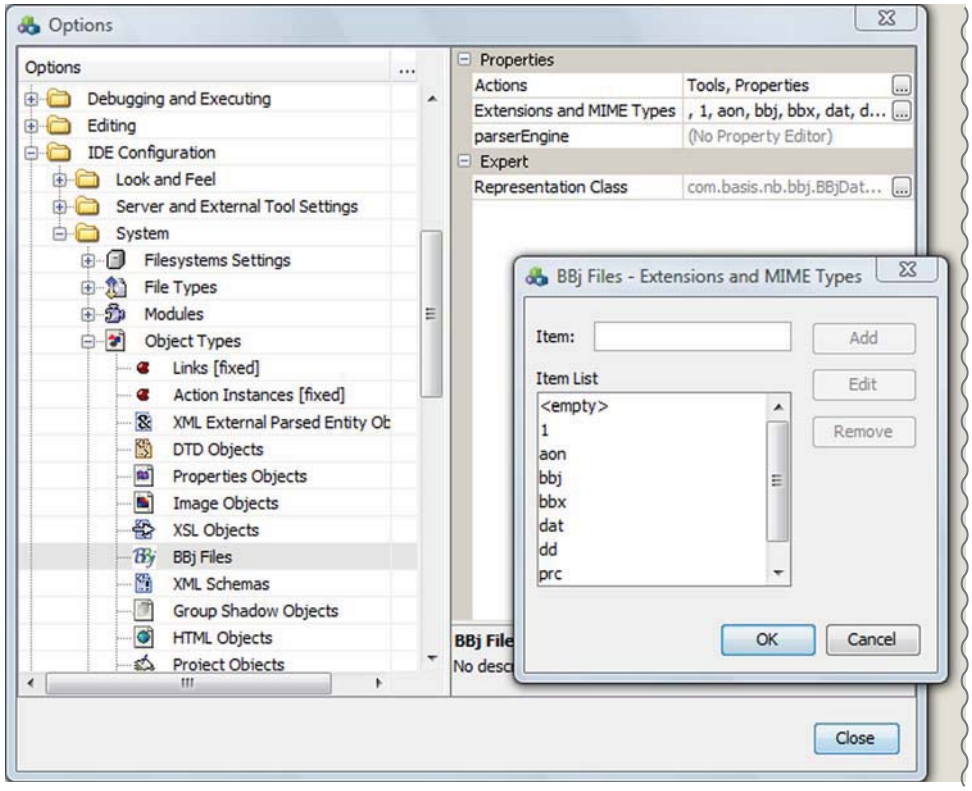


Figure 5. The Options window with the BBj Files – Extensions and Mime Types dialog

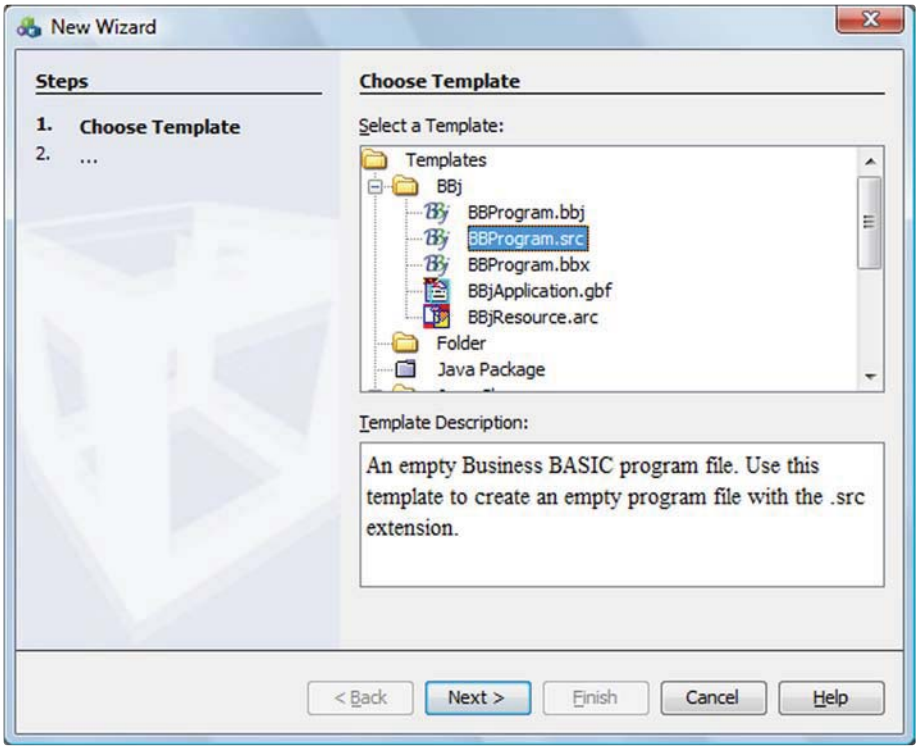


Figure 6. The New Wizard showing the BBj file templates

Would you like every program source file that your company generates to have a copyright notice at the top? Or would it be handy to have some standard boilerplate code in every new file you open, or to have empty file templates that use other file extensions? You can add your own templates to the list shown in the New Wizard.

First, prepare a file containing whatever baseline text you would like included in your template, then give it a unique name and a file extension that all new examples made from the template should have. (Files meant to be used in FormBuilder or AppBuilder must have the .arc or .gbf extension, but program source files meant to be opened in the Source Editor can have any file extension that you have registered in the Options window Object Types, as we've already mentioned.)

When your file is ready, right-click on it in the Explorer and select **Save As Template…** This opens a dialog that lets you choose the folder (or in other words, the file type category) where your template will appear. Click [OK] to create a template based on your file. You can make changes to this template by going to the Options window and expanding Source Creation and **Management > Template**s. It is even possible to insert macro objects into a template, which can do interesting things like automatically inserting the current date and time or the name of the user who created the new file from the template.

Part II - Editing Source Code

You'll probably spend most of your time in the IDE editing your application's source files in the Source Editor. Think of the Source Editor as a word processor that is purposely designed to assist in writing computer code. It is outfitted with special features to help you work faster and avoid mistakes.

Syntax highlighting, where different types of BBx language entities are assigned different colors, fonts, or styles, helps you discern the structure of your code. Verbs, functions, keywords, strings, comments, and other entity types are each assigned their own color, font, and style to distinguish them from one another.

All the attributes in **Figure 7** are customizable to suit your personal preference. Open the Options window and expand **Editing > Editor Settings > BB Source Editor/Debugger**, then click on the property editor (ellipsis box) of the "Fonts and Colors" property to display the Fonts and Colors dialog.

If you are writing BBj object-oriented programs that take advantage of the BBj API, code completion might become your best friend and constant companion. As the name implies, code completion can automatically complete or fill-in the statement you are currently writing in the editor. When you type certain "trigger" characters such as the period ('.'), the IDE reviews all the text in your file and all the files it refers to, in a process we call *parsing*. This allows the editor to make an educated guess about what you will probably need to type next. > >

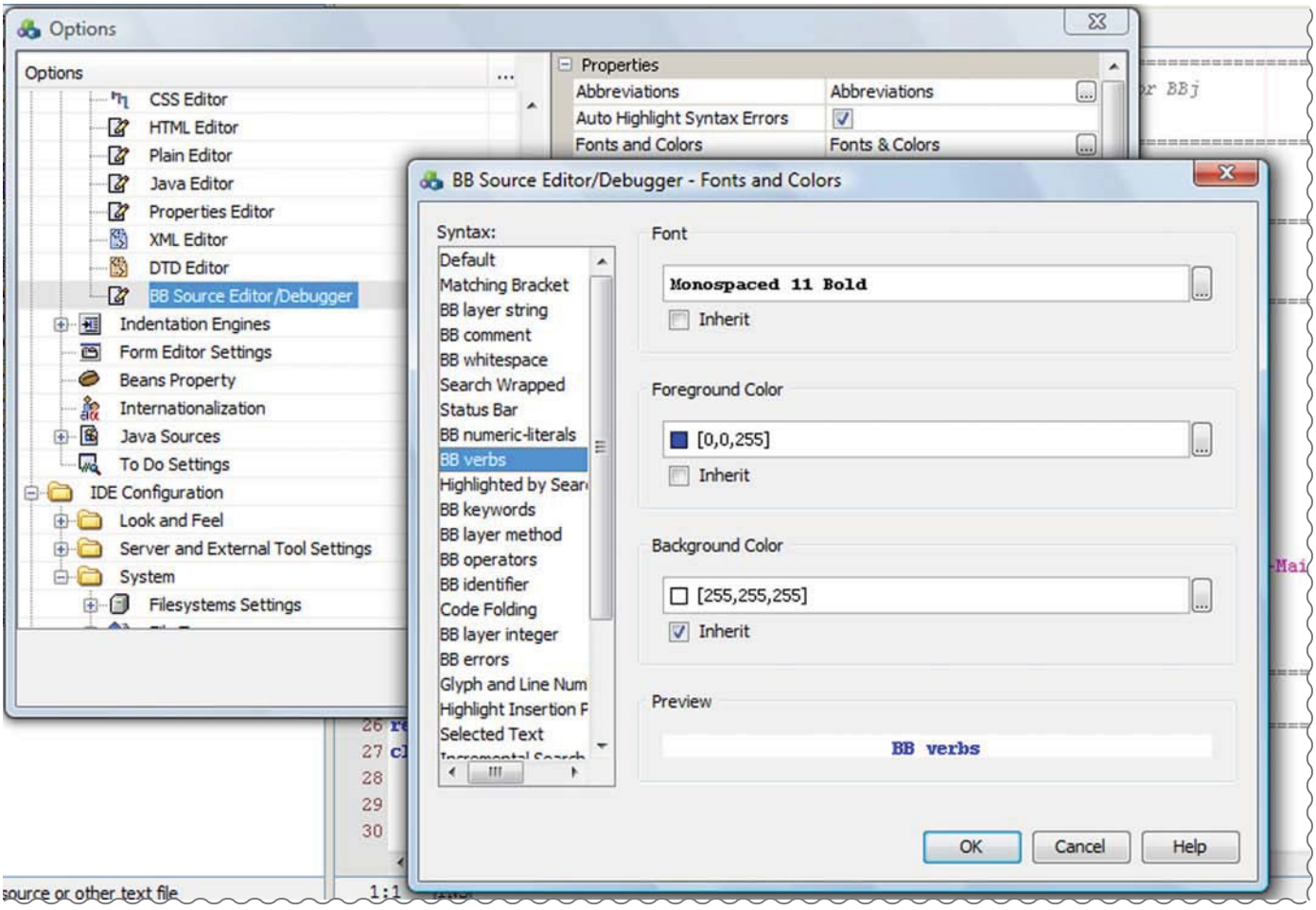


Figure 7. The Options window open to Editor Settings with the BB Source Editor/Debugger – Fonts and Colors dialog

Figure 8 shows a code completion popup window listing the possible choices that would be valid for an instance of the custom object class BBImageManipulator. Selecting one of these choices with the mouse, or by scrolling through the list with the up/down arrow keys and pressing ENTER, causes that choice to be inserted in your text at the cursor position. Code completion spares you the trouble of constantly referring to outside documentation to find out what methods or variables are available to a given class object, and also ensures correct syntax by preventing spelling or capitalization errors.

The Options window at **Editing > Editor Settings > BB Source Editor/ Debugger** offers two properties that affect code completion (see **Figure 9**). You can turn code completion off by unchecking the Auto Popup Completion Window property. If you would like a longer delay before the code completion window pops up, increase the time shown in the Delay of Completion Window Auto Popup property.

As your application grows and the number of program source files proliferates, finding your way around in them becomes more difficult. It is impossible to remember the exact location of every specific subroutine or algorithm. There needs to be a way to quickly identify and move to a section of code in any given file without spending a lot of time scrolling up and down in the editor, or searching for patterns that may or may not produce the correct results. > >

```
52
53      REM New way using the BBjScaledImage object
54      declare BBImageManipulator myBBImageManipulator!
55      myBBImageManipulator! = new BBImageManipulator(imageFile$)
56      myBBImageManipulator!.
57
58      REM Still not working
59      REM myBBImageManipulat
60
61      myButton2!.setImage(my
62
63      REM Print out the orgi
64      labe2! = myWindow!.add
65 :            + str(myBBImageMa
66 :            + "," + str(myBBImageManipulator!.getHeight()) + ")")
67
```

```
C:/dev/src/utilities/launchdock/bbimagemanipulator.bbj.BBImageManipulator
BBjNumber exists()
BBjNumber getHeight()
BBjImage getScaledImage(BBjNumber p_percentage)
BBjImage getScaledImage(BBjNumber p_width, BBjNumber p_height)
BBjImage getScaledImagePreserveAspectRatio(BBjNumber p_size, BBjNumb
BBjNumber getWidth()
BBjNumber gettransparency()
void settransparency(BBjNumber param!)
```

Figure 8. The code completion popup window

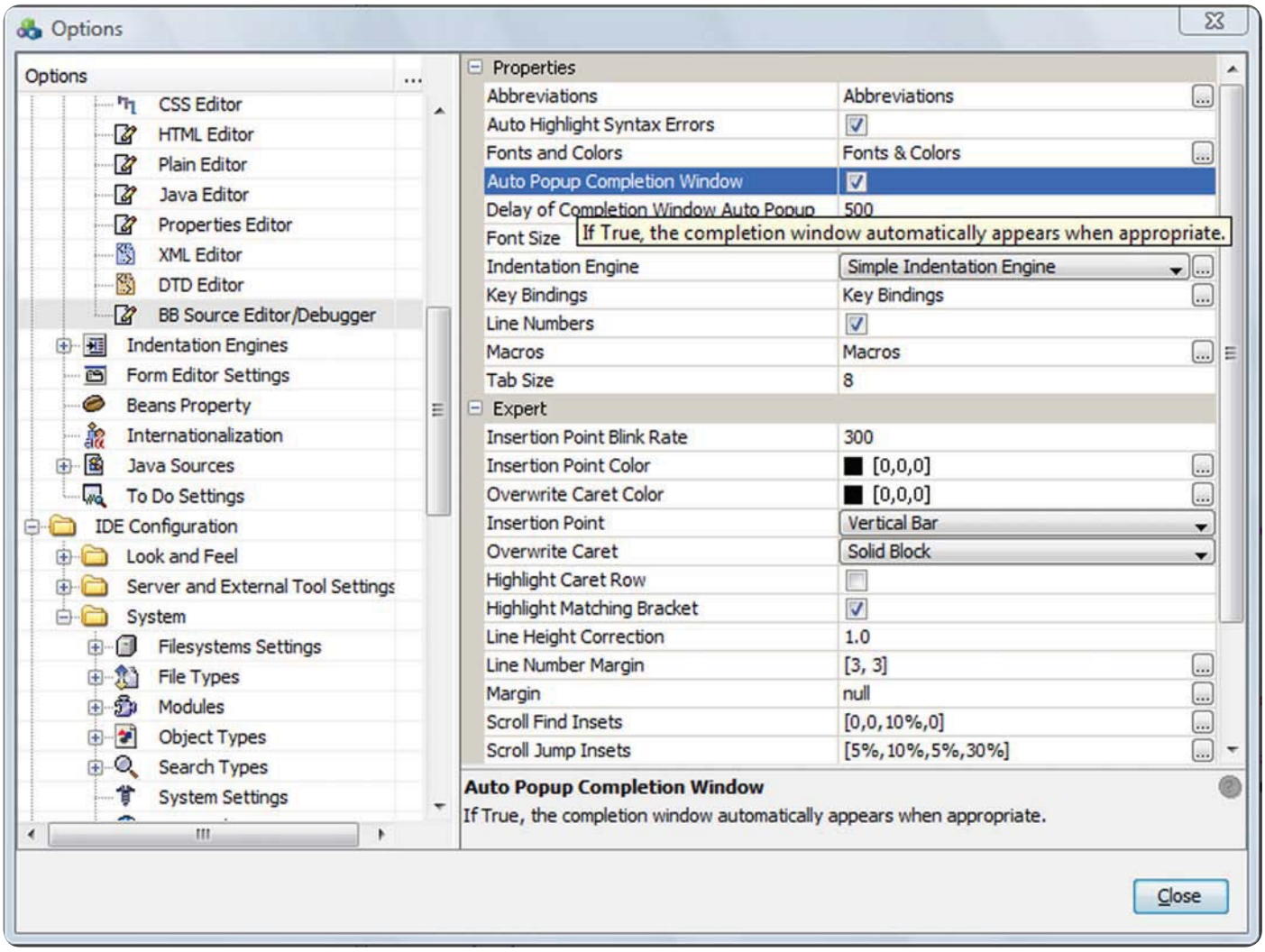


Figure 9. The Options window displaying BB Source Editor/Debugger properties affecting code completion

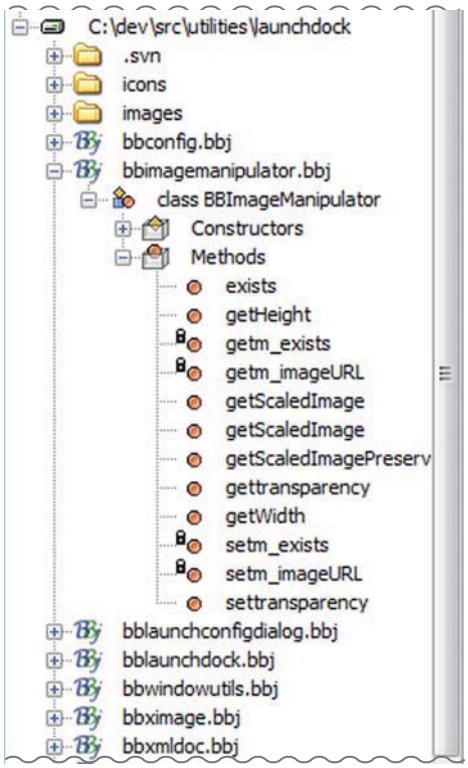


Figure 10. The Explorer showing an expanded BBj source file

The solution is the navigation capability built into the IDE's Explorer Filesystems tab, where you see a hierarchical view of your directories and files. As you would naturally expect, clicking on the little icon just to the left of a directory name causes it to expand and display a list of the files and subdirectories it contains. Did you know that program source files can also be expanded?

Figure 10 displays a file called *bbimagemanipulator.bbj* and all the classes, methods and subroutine labels it contains (which are called "child nodes" of the file they belong to).

This file isn't currently opened in the Source Editor, but double-clicking on one of these child nodes will have two effects:
1) the file will open in the Source Editor, and 2) the cursor will be placed at the start of the class or method or label that was selected. Expanding files in the Explorer and double-clicking their child nodes instantly moves the Source Editor to the corresponding section of code, without your having to do any time-wasting searching.

The parsing process, which updates code completion information and the child nodes used for navigation, also enables the editor to display syntax errors. Each time the file is parsed, all the syntax errors in the file you are currently editing will be highlighted. A simple way to force reparsing of your file at any time is to press [Ctrl]+Space. If you find this syntax error highlighting distracting and would prefer not to be constantly reminded of your mistakes, you can turn it off by unchecking the Auto Highlight Syntax Errors property in the Options window's **Editing > Editor Settings > BB Source Editor/ Debugger** page.

An even better way to catch syntax errors before attempting to run your code is do a test compile without writing any output. **Figure 11** shows the **Building > BBj Compiler Settings** page of the Options window. When the 'Compile Without Output' property is checked and the Error Log File property is left blank, no tokenized output file will be created and no error log file will be written to disk when you compile your source code.

When you compile your code, an output window opens at the bottom of the IDE that contains the list of errors that otherwise would have gone to the error log. Each error message is hyperlinked to the source file >>

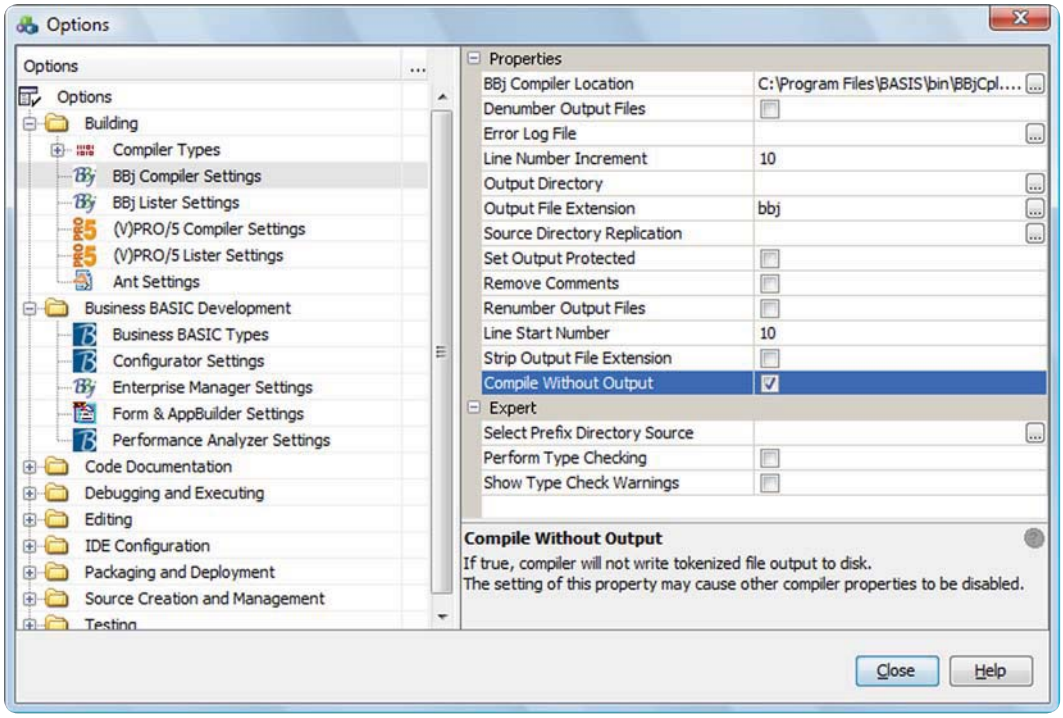


Figure 11. The Options window displaying the BBj Compiler Settings

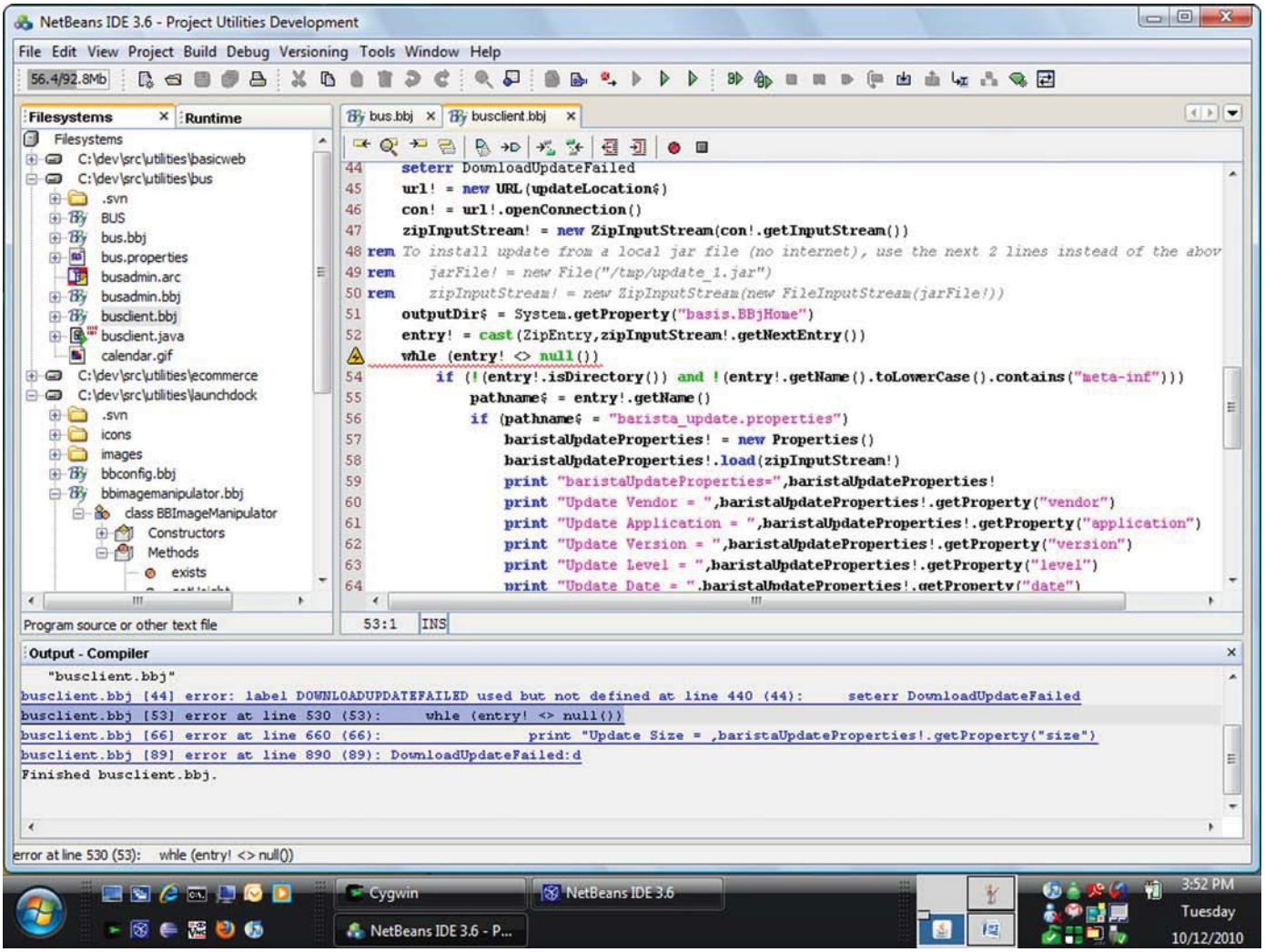


Figure 12. Syntax error hyperlinking

containing the error as shown in **Figure 12.** Double-clicking on a syntax error message opens the file in the Source Editor and positions the cursor on the line with the error.

Now that is a lot of what the IDE is capable of and how it can help you manage your projects and facilitate application development, but the IDE is extremely full-featured and can do ever so much more! Read on to explore more new features that improve the development experience with time saving techniques and powerful new capabilities.

Part III - Compiling and Deploying

Compiling or *tokenizing* is the process of turning ASCII text source code into a binary format that can be executed by the Business BASIC interpreter. It is one of the final steps before turning your application over to your customers and users, and something you will want to do often during development for testing purposes.

As a PRO/5 or BBj developer, you are already familiar with the pro5cpl and BBjCpl compiler utilities and their various command line options. The BASIS IDE provides a convenient graphical front end for these utilities, which simplifies the process of selecting and compiling large numbers of files. The IDE itself doesn't actually contain any built-in compilers; it merely connects to the cpl compilers found in your installed version of BBj or BBx. At compile time, the IDE assembles all the selected source files, invokes the specified cpl compiler as a process in a new execution thread, feeds it the appropriate command line parameters garnered from the properties you have specified in the Options dialog, and then displays the results when the process is finished. Using the IDE for compiling spares you a lot of careful typing and shows you what's happening in a more comprehensible way. As we already mentioned in *Part 2 Editing Source Code*, any errors found during compilation are displayed in the Compiler Output Window as hyperlinks. Double-clicking on a hyperlink opens the original source file in the Source Editor, where the line containing the error is highlighted. > >

Open the Options window and expand **Building > BBj Compiler Settings** to display a list of the properties for compiling with BBjCpl, as shown in **Figure 13**.

Most of the properties you see here directly correspond to parameters you would enter on the command line if you were using the compiler in a shell outside of the IDE. Holding the mouse cursor over the name of the property causes a "hint" window to appear with a short explanation of what the property controls. Most of these properties are obvious, so we'll spend some time looking at the most critical or unusual ones.

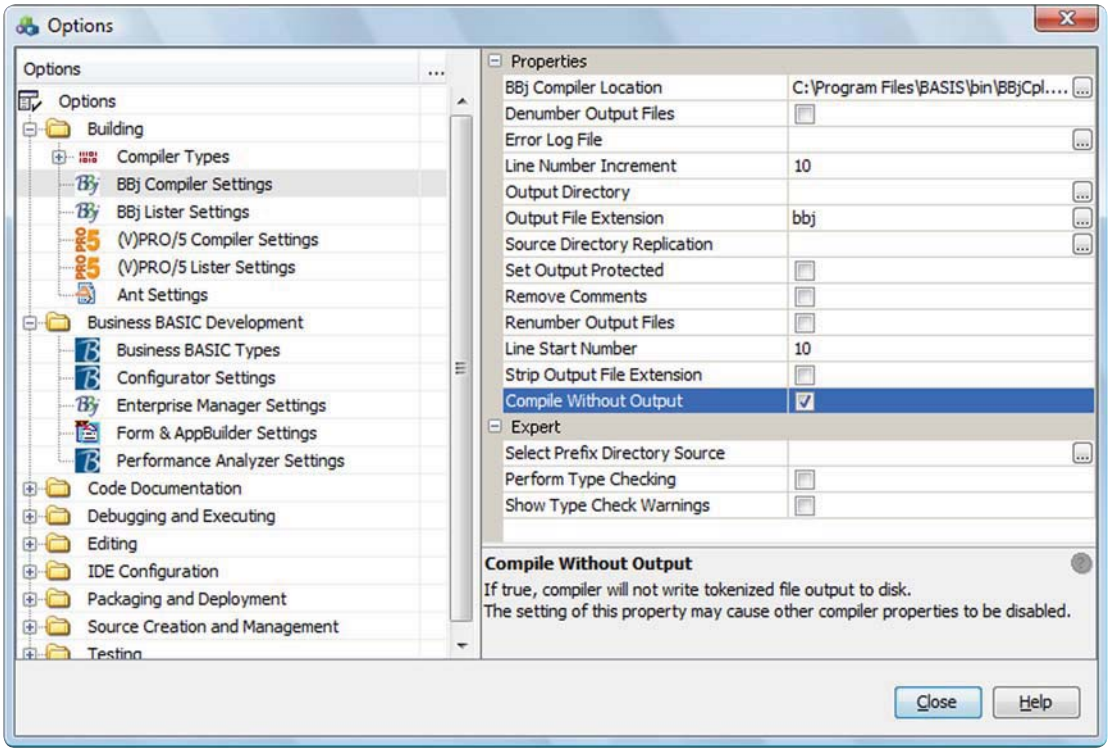


Figure 13. The Options window with the BBj compiler options.

BBj Compiler Location – This property comes pre-configured with BBj. It simply points to the installed BBjCpl compiler. If you move the utilities around or store them in unusual places, you will need to update this property.

Error Log File – This property corresponds to the **-e** command line parameter. It is worth mentioning only because if you specify an error log file, the compiler errors get written to a disk file instead of displaying in the IDE's Compiler Output Window. You must leave this property blank in order to display compiler errors as hyperlinks connected to the original source.

Output Directory – This of course represents the **-d** command line parameter and specifies the path to the directory where the compiler places the tokenized files. For best results, you should have previously created and mounted the output directory in the NetBeans Explorer. If this directory does not yet exist, the BBjCpl compiler will create and populate it with tokenized files, but you won't see the results in the Explorer Filesystems tab and may wonder if anything really happened. To create tokenized files in the same directory as the source files, leave this property blank.

Output File Extension – This is the **-x** command line parameter, and specifies the extension added to the end of tokenized files. The extension is not limited to a certain number of characters. If the extension is registered as a BBj module extension, the compiled files are assigned a special icon in the Explorer that distinguishes them from all other file types. Leaving this option blank triggers the file naming behavior discussed in the 'Strip Output File Extension' property below.

Source Directory Replication – This property has no command line equivalent; it represents a capability added by the IDE that you can't easily get with plain vanilla BBjCpl at the command prompt. This property represents the name of a specific directory from the directory path of the source file(s), which becomes the point at which source directories are replicated in the output directory structure. It modifies the directory path string that is forwarded to the compiler with the **-d** parameter so that the generated output files will exist in a directory tree hierarchy which is similar to the directory tree of the source files. Did you get all that? If not, don't worry. We're going to discuss this in more detail a little later.

This property is not essential for compiling; it can be safely ignored if you have no special concerns about the default handling of the output directory. If no output directory is specified in the 'Output Directory' property, this property is ignored because your compiled files will be created right alongside your source files in the same directory structure.

Strip Output File Extension – This property also has no corresponding BBjCpl command line parameter. When set to 'true' (when the box is checked), it removes the last file extension from the name of the compiled output file, provided that an output directory is specified and the 'Output File Extension' property is left blank. If no output directory is entered in the 'Output Directory' property, the 'Strip Output File Extension' property is ignored and has no effect on the naming of the file. To make this perfectly clear, here's a summary of the rules applying to tokenized file names and extensions: **> >**

Case 1: Output file extension supplied.
The tokenized file is named with the specified extension. The 'Output Directory' and 'Strip Output File Extension' properties do not affect file naming.

Case 2: No output file extension supplied. No output directory supplied.
The tokenized file is located in the same directory as the source file and has the same name as the source file, minus the last extension of the source file.

Case 3: No output file extension supplied. Output directory specified. Strip Output File Extension set to 'false' (unchecked).
The tokenized file is placed in the output directory, and has the same name and file extension as the source file.

Case 4: No output file extension supplied. Output directory specified. 'Strip Output File Extension' set to 'true' (checked).
The tokenized file is placed in the output directory, and has the same name as the source file, minus the last extension of the source file.

Compile Without Output – We touched on this property earlier in our discussion of editing source files, but this time the setting needs to be different. You are compiling source files with the intent of getting tokenized files written to the hard disk, so this time the property needs to be 'false' (unchecked).

Select Prefix Directory Source – This property represents the **-P** or **-c** command line parameters, depending on which of the four possible directory prefix list sources is selected. A prefix directory list is necessary when the compiler is directed to perform type checking of BBj object syntax, and therefore this property is ignored unless the 'Perform Type Checking' property is set to 'true'. The four possible settings are shown in the BBj Compiler Settings – Select Prefix Directory Source dialog in **Figure 14**.

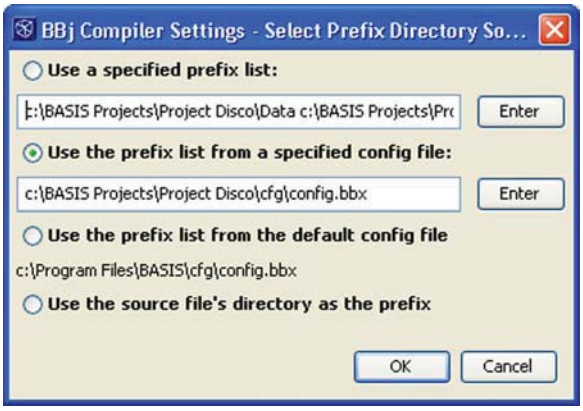


Figure 14. The Prefix Directory Source dialog

Use a specified prefix list. Select the radio button, type a space-separated list of directories in the text field and press the Enter button. The command line arguments forwarded to the compiler will include the **-P** parameter and the list of prefix directories from this text field.

Use the prefix list from a specified config file. Select the radio button, type the path/file name of the config file that contains the desired prefix directories and press the Enter button. The command line arguments forwarded to the compiler will include the **-c** parameter and the path/file name of the config file from this text field.

Use the prefix list from the default config file. Select the radio button. The command line arguments forwarded to the compiler will include the **-c** parameter and the path/file name of the config file specified in the BBj Execution and Debug Settings > Config File Location property.

Use the source file's directory as the prefix. Select the radio button. No **-P** or **-c** parameters are added to the command line forwarded to the compiler. If type checking is required (the 'Perform Type Checking' property is set to 'true'), the compiler will use the source file's directory as the directory prefix.

Perform Type Checking – This represents the **-t** command line parameter. When set to 'true', the compiler will perform type checking on BBj object syntax used in the source file.

Show Type Check Warnings – This is the **-W** command line parameter, which is ignored unless the 'Perform Type Checking' property is set to 'true'. When set to 'true', the compiler will display type check warnings as well as errors in the output results.

We ought to mention that certain BBjCpl command line parameters are not available when compiling BBj source code with the IDE. These include the ? parameter, which displays a usage summary, the @ parameter for specifying a text file containing a list of files to be compiled, and the **-R** parameter for recursive compiling of files in subdirectories. (As we will soon find out, in the IDE recursive compiling is performed by selecting a directory instead of a file in the NetBeans Explorer and choosing **Compile All** or **Build All** rather than **Compile** or **Build**.)

Now that we've covered the most important compiler options and you are all configured, let's get on with the actual compiling (the hard part is over). To compile a file in the IDE:

1. Select a program source file in the Explorer Filesystems tab.
2. Right click to open the popup menu (or open Build on the main menu).
3. Select **Compil**e or **Build** (or use their short-cut keys) to kick off the process.

A quick word about the difference between '**Compile**' and '**Build**': When you select **Compile**, the IDE checks to see if a compiled version of the file already exists, and if the date/time of its creation or last modification is more recent than the last modification date/time of the source file. This is called the "up-to-date" check. If a compiled version already exists and is up-to-date, the source file is not recompiled. The **Build** choice does not check for pre-existing and up-to-date compiled versions of the source. All selected source files are compiled again, regardless of the status of any already compiled versions. Choosing Compile instead of Build may therefore save you some time, since only the changed files in your selection are recompiled rather than all of them. > >

You can compile more than one file at a time by holding down the [Ctrl] key and selecting multiple files, or you can select a directory in the Explorer instead of a file. Here's where we talk about recursive compiling, which is merely the difference between **Compile/Build** and **Compile All/Build All**. When you select a file or group of files in the Explorer, you are offered the choice to **Compile** or **Build**. If you select a directory rather than a file, the choices expand to include **Compile All** and **Build All**. The **Compile** and **Build** options work only with the files in the selected directory. Compile All and Build All process not only the files in the selected directory, but also the files in any subdirectories of the selected directory.

The **Clean** and **Clean All** menu choices (as you might expect) are there to wipe out the tokenized files if you no longer want them. **Clean** will delete only the files in the currently selected directory, while **Clean All** does a recursive delete through all the subdirectories of the selected directory. Use with care!

As promised, we need to step you through a more detailed description of the 'Source Directory Replication' compiler property and discuss its implications. Source directory replication is only going to be interesting if you keep your source text files in a different place than your tokenized files. If your preferred development style is to lump all the hundreds or thousands of files in the entire application into the same directory, you don't have any source directory structure to replicate and you should leave this property blank or set it to 'Use no source directory replication at all'. (However, if your preferred development style is to lump all the hundreds or thousands of files in the entire application into the same directory, we'd love to convince you there's a Better Way To Do Things.)

The 'Output Directory' property of the compiler options lets you set the destination of the generated tokenized output files. If you leave this property blank, the compiled files are created in the same directory as their corresponding source. If the 'Output Directory' property is filled in with a directory path, the IDE automatically derives a new output path based on the 'Output Directory' property and the location of the source files selected for compiling. This new output path is used as the **-d** parameter given to the cpl compiler, as shown in **Example A** below.

Example A
Setting of the Output Directory property:
```
C:\BBj Development\level A\level B\bin
```
Directory containing source files:
```
C:\BBj Development\level A\level B\src\level C\level D
```
Adjusted –d parameter sent to the compiler:
```
C:\BBj Development\level A\level B\bin\level C\level D
```

In this example, the source files are kept under **src** and the tokenized binary files under **bin**. The IDE computes the directory path for the **-d** command line parameter by comparing each segment of the source path with each segment of the output directory, starting from the beginning of the path. Each segment of the source path is identical with each segment of the output path until the **src** segment, so all segments after src in the source path are appended to the output path, which is then used for the **-d** parameter. This automatic path derivation algorithm insures the output from files in src will go in **bin**, the output from src\level C will go in **bin\level C**, the output from **src\level C\level D** will go in **bin\level C\level D**, etc. The automatic path derivation is in effect when the 'Source Directory Replication' property is set to 'Use default source directory replication'.

In **Example B**, this automatic derivation algorithm fails to make the expected output path:

Example B
Setting of the Output Directory property:
```
D:\Our Product\Release 4
```
Directory containing source files:
```
C:\Development\Our Product\beta\gui\main menu
```
Incorrectly adjusted –d parameter sent to the compiler:
```
D:\Our Product\Release 4\Development\Our Product\beta\gui\main menu
```
>>

The first non-matching segment is **C:**, so everything after the first segment is appended to the output path specified in the Output Directory property. This results in a long, awkward output path that probably isn't what was desired. The 'Source Directory Replication' property's 'Use a selected directory for replication' setting is designed to correct this problem and help the output path derivation algorithm do the right thing, as shown in **Example C**:

Example C
Setting of the Output Directory property:
```
D:\Our Product\Release 4
```
Directory containing source files:
```
C:\Development\Our Product\beta\gui\main menu
```
Setting of the Source Directory Replication property:
```
gui
```
Correctly adjusted –d parameter sent to the compiler:
```
D:\Our Product\Release 4\gui\main menu
```

Because a segment from the source directory path is identified in the 'Source Directory Replication' property, the automatic derivation algorithm can skip the segment comparison process. Everything in the source path starting with the **\gui** segment is appended to the output directory, which results in a much more appropriate **-d** parameter output path.

And so you reach the end of this rather long IDE discussion, we hope you find it helpful! ■

Download the latest version of BBj and enjoy these new enhancements:

Updated for Java 1.6:
- Enhanced NetBeans 3.6 codebase no longer requires Java 1.5 to run; now uses the same Java 1.6 version as the BBj server.
- Adds new Java 1.6-compatible code completion database for Java code development.

Enhanced code completion:
- Adds file navigation from the Explorer. Expand file nodes to see classes, methods and labels contained in the file; click on one of them to zoom to that spot in the file.
- Includes new BBjConstants; a BBj API class containing often-used constants for constructs like MSGBOX, making them accessible through code completion.

Have You Missed an Issue?

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

Partnership

2008	#1	Starwood's Web Service Shines Brighter
2007	#1	RCG Uses Marketing Skills to Achieve OSAS "Top Dog" Status
		Marex Returns Home to BASIS – A Personal Journey
2006	#1	Borealis Press Chooses Mac Platform
		BASIS Tours Europe With New Partnership Program
2005	#2	OSI Solidifies OSAS Investment with BBj
	#1	BBx Generations – Our Progression in Time
		MARK SYSTEMS – A Revenue Stream Runs Through It
		Enhancing the Bouquet with BBj – KMK Software AG
2004	#1	Fit For the Future? – Midata Service GmbH
		BBj Sparks High Productivity at Heilind Electronics
2003	Q4	BASIS Introduces Online Tutorials
		Maharam: BASIS Customer Profile
		Technical Resources on the BASIS Web Site
	Q2	SAM Plan – Managing Your Software Assets

Language / Interpreter

2009	#1	MDI Modes: Make Mine a Single, With a Twist
		Jetty Offers Legacy Programs via Web Services
		From Legacy to Enterprise with BBj Web Services
		Object-oriented Performance
		A Sneak Peek at BBj's Browser User Interface
		The BASIS Java Hat Trick
2008	#1	Dishing up Forms
		T Minus - BASIS LaunchDock Blasts Off
		Blowing the Doors Wide Open With ClientsObjects
		Quick and Easy Solutions With Java Libraries - Part 1
		Quick and Easy Solutions With Java Libraries - Part 2
		BASIS Adds Mapping Functionality to BBjRecordSets
		More on Client File Access
		Inner Types Exposed
		Socket to Me
		Deliver a Package Tracking System
		Real-time Web Reports
2007	#1	Spin, Slide, or View Your Data
		If You Have Choices, We Have Choosers
		BBj Spell Checker is all the Buzz
		A Tour of the BBjCharts API
		Leap from 'Cut and Paste' to 'Drag and Drop'
		Desktop Data Delivered
		Catching the XML Wave
		A Deeper Voyage Into the BBjCharts API
		TRZ: Jars, Jars, and More Jars
2006	#1	A Primer for Using BBj Custom Objects
		Confessions of a Language Polygamist
		Applying Custom Objects to Existing Code
		The Scoop on 64-Bit Computing
		Visual PRO/5 6.0 with an XP or Vista L & F
		Type Checking With "bbjcpl"
2005	#2	Grid to Order
		Safe and Retrievable Intellectual Property
	#1	Input Validation: Veto Power
		Connection Pooling – Part 1
		Connection Pooling – Part 2
2004	#1	The Secrets of Secure Communication
		Why Use the BBjRecordSet?
2003	Q4	BBj Databound Control
		BBj Form Printing
		GUI Date Input With the BBj INPUTD Control
		Using the BBjRecordSet
	Q2	New Language Features Give Programmers More Choices
		Writing a Web Service in BBj
2001	Q2	From BBx to BBj
		The GUI Enhancements of BBj

Database Management System

2009	#1	Database Query Analysis
		Online Copy Jobs
		BASIS Generates SPROC Template Code
		Debugging SPROCs and Triggers
2008	#1	Databound Grid - Data Abstraction Extended
		Take the Plunge Into SQL-Transactions
		Data Warehousing II
2007	#1	Unleashing the Power of SPROCs Without SQL
		Solving the Data Warehousing Dilemma
2006	#1	Using Triggers to Maintain Database Integrity
		Using Stored Procedures to Add Business Logic to the Database
		ESQL Files: Constraining Your Data to Guarantee Integrity
2005	#2	PDF Now Also Means Perfectly Displayed Forms
		OpenOffice.org – a Sweet New Suite
		VKEYED Files – New Features for a New File Type
	#1	The BASIS DBMS – New 5.0 Features
2004	#1	Secure Your Future, Encrypt Your Data
		The Great Connection
2003	Q2	BASIS International Offers a Choice of Databases
		Capitalizing on BASIS's Open Architecture
2001	Q2	Next Generation File Format: XKEYED Files

Development Tools

2009	#1	Barista Uses Your RDBMS of Choice
2008	#1	From the BASIS DD to a Barista App in a Flash
		From the BASIS DD to a Barista App in a Flash Part 2
		Goodbye Event Loop, Hello Process Events
2007	#1	Watch the Form Gen Wizard Trans"form" Data
2006	#1	AppBuilder: The BASIS IDE Gets RAD GUI Development Integration
		Freedom of Choice: Using Object Code Completion in the IDE
		BASIS Puts Triggers to Work
2005	#2	Tuning the Performance Analyzer
		The State of the IDE
	#1	FormBuilder: BASIS IDE's Better Cross-Platform Resource Builder
		TRZ: Logs, Logs, and More Logs
2004	#1	Inside BASIS: Software Product Version Control
2003	Q4	Managing Databases/Data Dictionaries with the BASIS IDE
		Viewing Data Files in the BASIS IDE
	Q2	Speeding Up the Development Process – The BASIS IDE

System Administration

2009	#1	Jetty Offers Legacy Programs via Web Services
		Jetty Web Server for BBj, Java Style
2008	#1	The Extreme Enterprise Manager Makeover
2006	#1	Solving the Locked Record 'Whodunit'
		TRZ: The FAQ's About 64-Bit BASIS Products
		Gain the Advantage – Use the Advantage Resource
2005	#2	Choices, Choices, Choices
		The Response File Responds Silently
		Safe and Retrievable Intellectual Property
		TRZ: Lock, Lock, Who's Got the Lock?
	#1	New Install Puts You in the Driver's Seat
		Taking Out The Garbage
2004	#1	Give Your Software a New Lease on Life
		TRZ: Licenses and BLMs
		Without a Trace: The Silent Installation

Building Blocks

2009	#1	A Friendlier 'AddonSoftware by Barista'
		Recipes for Successful Report Writing
		Jump on the BUS - BASIS Update Service
		Jazz up Your Applications
		Can Your App Speak to Your Customer?
2008	#1	Freshly Brewed - Barista Caffeinates Addon Accounting Applications!
		BBj Thrives on the Web

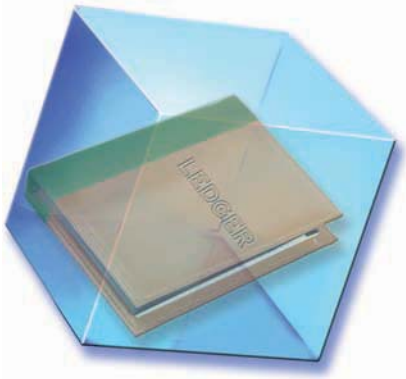
AddonSoftware® Partner Program

Delivering Building Blocks for Your Vertical Solutions

The **AddonSoftware® ERP** solution provides you with the building blocks through its core accounting, distribution, and manufacturing bundles to fully complement your vertical market.
Don't reinvent the wheel!

Simpler

- Scalable pricing, product, and margins
- Access to source code
- OS independent

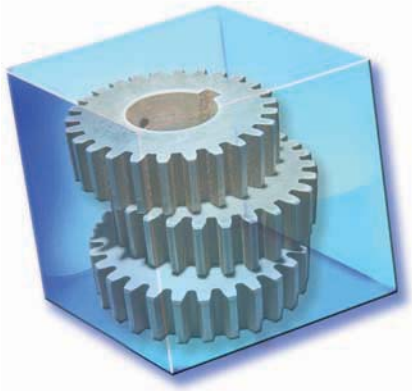


Faster

- Easy customization with call point code
- Includes the Barista RAD tool

Integrated

- Out-of-the-box functionality
- No dependencies on third party utilities
- Deployment flexibility



Unique

- Preserves your customizations through upgrades
- Co-operative development community with your choice of involvement

Interested?

Join the growing AddonSoftware community today to make the most of your expertise while maximizing your revenue opportunities.
Contact us to find out more.

1.800.370.9131 U.S. and Canada • +1.505.338.4188 International • info@addonsoftware.com • www.addonsoftware.com