$7.00 U.S.

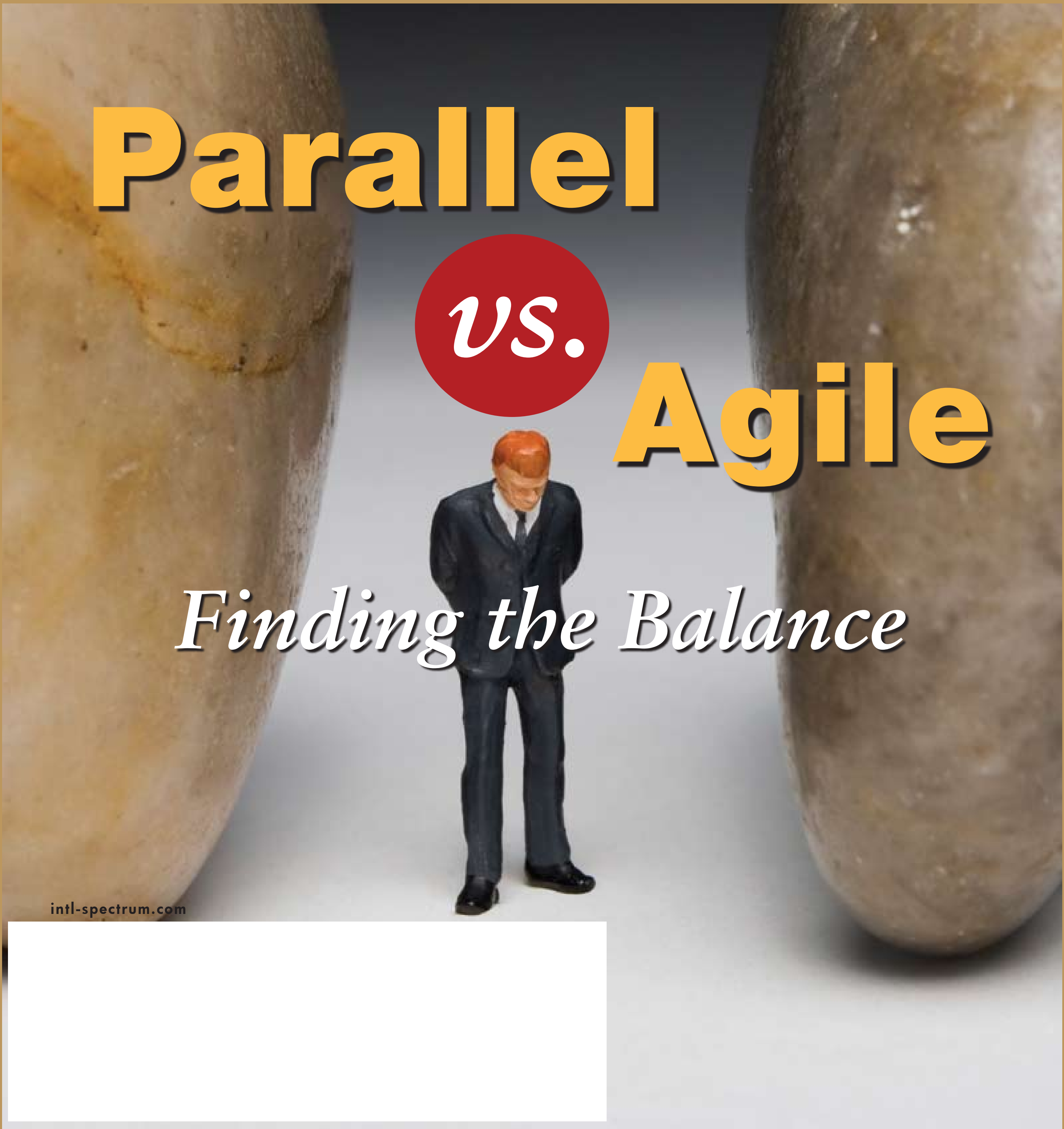
INSIDE: Predictions for 2012

INTERNATIONAL

Spectrum

THE MULTIVALUE TECHNOLOGY MAGAZINE | JAN/FEB 2012

PLUS! jQuery Mobile 1.0 released



Breakthrough Development.

Develop breakthrough applications that *share data, provide insights,* and *drive actions.*
Develop with InterSystems. Instead of separate systems for connectivity, database, and analytics.

For a video and whitepaper about breakthrough applications, visit **InterSystems.com/Breakthrough1WW**

INTERNATIONAL Spectrum
THE MULTIVALUE TECHNOLOGY MAGAZINE

FEATURES | JANUARY/FEBRUARY 2012

DEPARTMENTS

From the Inside

Big Data has been a hot topic lately. Since it really is not clear what it is, or how to Big Data applies to us, I figured I'd add my two cents on the topic.

First of all, Big Data is not new. This problem has existed and been handled for much longer than the last year or so.

Big Data references to the complexity, amount, and the management of large quantity of data. If you look at the definition on Wikipedia, you will see it talks about "terabytes and petabytes" of data, but it also states "beyond the ability of commonly used software tools to manage and process within a tolerable time."

Big data is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target currently ranging from a few dozen terabytes to many petabytes of data in a single data set.

Reference: http://en.wikipedia.org/wiki/Big_data

Let's look at the concept of "tolerable time" and "software tools." At first glance you may say, well, I don't fall into that category. I can manage all my data in my programs within a "tolerable time."

"Tolerable time" is extremely relative due to the "who, when, and why" an enterprise has to deal with on a daily basis. Are you generating reports? Are you generating lookups and presentation tools for immediate action or batch processing? When does the user expect this information to be presented to them? Now? Or in 30 minutes?

The "Big" in big data is relative as well. I've seen articles on the Internet arguing about what constitutes "Big data sizes." Some define it in the amount of data being stored and the size of the database.

Let's look at the classic example of sensor data. This data is usually highly structured and not very big per transaction (50 Bytes). But there can be a large number transactions. If that is the case, one million transactions would be about 48 MB. That is not "terabytes and patabytes." It would take 21,990,232,555 transactions to reach one terabyte (60,247,212 transactions a day for one year). While this may happen in your environment, in a normal business environment this is not likely.

Working with Big data is not about the size of your database, but the complexity of the data you need to work with. The idea of "Big Data" was to address problem of working with your complex data to find trends and solve problems with large volume and variety of data.

So how does "Big Data" apply to us? For the most part, "Big Data" is more Hype and Marketing that anything real. Our business data deals with "Big Data" on a regular basis — just based on the sheer volume and complexity of the interrelated nature of your business systems.

If you want to hear more about "Big Data," join us at the 2012 International Spectrum Conference to see how people are handing their own "Big Data" and the different techniques you can use to handle your Big Data issues.

-NATHAN RECTOR

President, International Spectrum

nathan@intl-spectrum.com

twitter.com/intlspectrum

intl-spectrum.com/facebook

INTERNATIONAL Spectrum

JANUARY/FEBRUARY 2012

Learn more about the MultiValue Symbol and see what MulitValue Technologies and MultiValue Communities exist to help you support and manage your business and systems. To find out more visit

http://www.intl-spectrum.com

MISSION STATEMENT *International Spectrum* magazine's editorial mission is to be the premier independent source of useful information for users, developers, and resellers of MultiValue database management systems, open systems business database solutions, and related hardware, software, and peripherals. Published bimonthly, International Spectrum provides comprehensive coverage of the products, companies, and trends that shape the MultiValue marketplace as well as the computer industry at large — helping its readers get the most out of their business computer systems.

International Spectrum is published six (6) times per year at the subscription price of $40.00 U.S. in the U.S.A.; $45.00 U.S. in Canada and Mexico; $50.00 U.S. for other countries. Single copy rates are $7.00 U.S. in the U.S.A. and Canada, and $9.00 U.S. in all other countries. International Spectrum is published by International Spectrum, Inc., 8956 Fox Drive #102, Thornton, CO 80260; Tel: 720/259-1356; Fax: 603/250-0664 E-Mail: request@intl-spectrum.com.

PRINTED IN USA

NEWS RELEASES/UNSOLICITED ARTICLES
International Spectrum is eager to print your submissions of up-to- the-minute news and feature stories complementary to the MultiValue marketplace. Black and white or color photographs and diagrams are welcome. Although there is no guarantee a submitted article will be published, every article will be considered. Please send your press releases, articles, and queries to: editor@intl-spectrum.com. International Spectrum retains all reprint rights.

The Rock and the Hard Place

Parallel and Agile Development

Part 1: Parallel Development

BY SUSAN JOSYLN

***Note about the series:** As we are reach for the best way to deliver the highest quality of software and services, we find ourselves considering parallel development and pondering the whole concept of agile. It may be that there is a sweet-spot between the two, and for each company that might be a slightly different spot. These articles will explore the perils of parallel development in part one, the angst of agile in part two, and how to pick and choose to find the right balance for your organization in part three.*

Achieving preposterous results with negligible resources is simply an IT standard these days. We are developing across multiple platforms, industries, and teams which often include offsite or even offshore members. Yet we must bring it all together — cohesively and quickly. Our challenge is made more difficult by the common IT realities that author Johanna Rothman[1] identifies as "Pants on Fire" and "Split Focus" management. Familiar to most of us, "Pants on Fire," describes how IT resources are rushed from one objective to another, never able to stay on any one until it is "done right." "Split Focus" is what we end up with when, in an effort to overcome this chaos, objectives are parceled out in portions to the team.

> *Companies have grown frustrated with the waiting game of "serial development." The whole idea of coding something then sitting on our hands for it to be tested and accepted before coding the next thing is "old school."*

A team may be admonished to spend "half" of their time on this and to juggle three or four other projects in some precise proportion. Since neither of those methods is making stars, the natural logical progression is to split the development into teams. The idea is that each team (or individual) has the luxury of remaining entirely focused on their project until it is (somewhat) completed. In reality this leads to yet another management strategy we can call "Herding Ducks." Once you let these teams and the software scatter off in all directions how will you ever gather them together again?

As scary as we may find it, the lure of parallel development is powerful even for single team environments. Even in that scenario we may be developing for a future release but find that we must make a "quick fix" to a version that's already in the field. We need access to a copy of the source code for that released version — without disrupting (or pre-delivering!) the new code that's under development. So we make our change to the field-release copy and get it to the customer quickly with no interference. Now that fix has to be reintegrated into what is coming from development or it will be lost and the bug will be reintroduced. Even in

[1]Johanna Rothman, author of "Pants on Fire or Split Focus" published on Tech Well (www.techwell.com).

Digital Version On-Line

Scan Me!

small-volume, slow-moving environments (and find one of those!) difficulties arise with conflicts and bottlenecks. Taking that to the next level where we have teams making different changes to different code streams and we end up with a proliferation of branches that can cause developers to make inadvertent changes to the wrong code. Even if we don't, we have to remerge and retest everything repeatedly. Do we spend more time cleaning up than we saved? Advocates against parallel development are quick to point out that nine women cannot make a baby in one month. It takes one woman nine months and there is no way around this. For these naysayers, the very idea of having change occurring in more than one stream is a ticking time bomb.

"Branching is a distraction and serves to decrease productivity and code robustness." contends Steve Berczuk in his article *Branching to Distraction*.[2] Going back to that "fix," regardless of its nature, we are now splitting our attention from the current project work. We may be fixing a problem that has already been addressed on the active development line, and we are constrained to using versions of frameworks and tools with which the released code was delivered, even though using the newer versions of those would make the work easier. No matter what, developers have to deal with the frustration and cognitive costs of working with a code base that is different than what they work with daily. And that's just to make the change, before even thinking about how and what to merge after it is completed. Merging a complex change to a heavily changed main line runs the risk of becoming a branch in and of itself.

On the other hand, companies have grown frustrated with the waiting game of "serial development." The whole idea of coding something then sitting on our hands for it to be tested and accepted before coding the next thing is "old school." As a result, most of us are under pressure to consider parallel development despite our reservations. So let's consider it. Parallel development can be implemented in many ways but has in common the underlying idea that two or more "code steams" are being developed. The copies are referred to as branches and they also have in common that ultimately they must be merged. Beyond that the implementations can vary wildly. Here are some common practices.

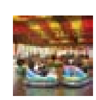
- **Just Say No.** Many organizations still forbid branching entirely. They've weighed the cost and benefits and decided against it. They do all development in one main trunk. If a fix *has* to be delivered to an old release, it is peeled out of the main development area and hand-integrated on an individual basis. This is a monumental effort for which sacrifices are made all around. It can be very costly in a ways that are not obvious. What often ends up happening is that programmers have their own "sandboxes" — separate work arenas. These isolated sandboxes are not sanctioned, integrated, nor automated. In actual practice, no doesn't mean no. It just means not managed.

- **No Merging.** Branching is accommodated in limited circumstances with *no* merging. You can split off development in some situations, but you never bring it back. For example, after a fix is made to a field release it is then opened as a new bug report for the new release and handled separately. Of course, these fixes have to have a high priority in the main trunk because they must be completed before the associated programs go live with the old bug still in place! This is practicable when there is truly minimal branching and the main trunk is released frequently (if you begin to hear the echoes of agile practices, you have good hearing.)

- **Critical fixes only — just wing it.** Because we don't branch very often, we don't have to plan for it. In a crisis, do whatever is necessary. This is a small step from "Just say no," so the branch management may be less hidden, but is no less damaging. Remarkably common, this method grows like weeds in your garden — genetically modified weeds that are nearly impossible to eradicate.

- **Cinderella Method.** Go off in your own little development corner and bring back something finished. This is cycle branching and merging where the merge is done at a set point in the life-cycle, often after release to production. It has a structure that can be planned for and accommodated with procedures and tools, but has its own inherent risks. Often work is lost on both sides of the divide. Time is always lost. Sometimes the work to merge the branches becomes its own branch. This multiplies the time lost exponentially.

- **Continuous integration.** A pillar of agile practices, the idea of frequent integration is taken to extreme heights with daily builds in some environments. Whatever the interval period, the idea is that we don't wait for projects to be completed, but rather do the merge work on a predetermined periodic basis. Frequent integration has the advantage of identifying integration issues quickly, but the disadvantage of requiring the person doing the merge to shift context frequently. Plus, if the idea was to not make the two developers interfere with each other, this is completely defeated — worse than defeated — by making them merge



Continues on page 8

[2]Steve Berczuk, author of "Branching to Distraction", published on Tech Well (www.techwell.com).

THE ROCK AND THE HARD PLACE: PARALLEL AND AGILE DEVELOPMENT — PART 1: PARALLEL DEVELOPMENT

Continued from page 7

into each other's work almost continuously! This is a particularly difficult concept in the MultiValue world. While a particular program is compiled — and this might be similar to a build in that it quickly reviews the syntax and we can make sure the program at least runs — most of our work to complete a software change requires changes to several types of items, and these are not "compiled" together into a unique, executable deliverable. The idea that you could run a "build" and make sure the whole application still works is ludicrous in our environment. The regression testing involved in this would be astronomically larger than the development effort. It's hard enough to get testing done as it is!

If your mind is already connecting the Cinderella-off-in-a-corner-until-its-done and the continuous integration approaches and seeing where identifying smaller tasks that are completed and integrated frequently could work, go to the head of the class. This is where parallel development meets agile on ground solid enough for MultiValue.

MultiValue Mechanics

There are several approaches to parallel development that are practiced in MultiValue. Each has some benefits and some risks. The one that will be best for any given environment depends on a lot of factors. In fact it would vary by project, by team, by day — perhaps with the weather.

Changing item names

Since our MultiValue environment is "granular," we can actually manage branch copies right in the same development arena (or account). If you make a change to a program that is already checked out to another developer, you can rename your copy. However, when you would like to test it you find that it is called from a menu that does some necessary pre-work so it cannot run standalone. You can't change the menu to call your copy so you make a copy of the menu, too. Then a subroutine that you modified requires that you also branch the calling program… and so on. You can end up with a proliferation of renamed items, many renamed just to call a renamed item for testing. Still, it is do-able and ultimately when you complete testing and are ready to merge your work back into the main trunk, those items will be easy to revert back. This method works best with frequent integration / small projects because probably multiple projects that change the same items should be tested and delivered together. If a major project is under development in the same arena it can make it very difficult to get the renamed copies past it — that situation lends itself better to separate arenas.

Changing file names

Some of the renaming just to call a renamed item can be avoided by leaving a program with the same name, but move it into a separate file. When you are ready to test you can catalog your version of the program. This method really only accommodates one branch being tested in the account at a time — and even worse, it is not at all obvious which version of the program is cataloged at any given moment.

Separate arenas (accounts)

The idea of creating a whole separate "development account" for each project begins to appeal. No renamed components, no mystery catalogs. Work any way that you need to work and then roll your work back into the main trunk. Space can be an issue, unfortunately, because assuming you already have a development copy of the environment, we are now talking about creating a second one — or more. These are whole, working copies of the environment — perhaps each one comprised of multiple MultiValue "accounts." Some companies like to have a whole copy for each programmer or team. We find in practice that this ends up being more complex than it needs to be — mainly because all of the accounts have to be updated with everything that is completed in all of the other accounts. (Picture braiding Rapunzel's hair.) The notion of separate arenas seems to work better when they are for a purpose, and for a set period of time, not for a person or team and not indefinitely. Having a main arena that is for ongoing fixes and small changes, then a separate one for some major development project can be a good compromise. If you can tip-toe your way toward agile practices with that bigger project — integrating it into the mainline at regular intervals — this is even better. The next article in this series will explore what agile practices are and how they translate to what we really do on a day-to-day basis. IS

SUSAN JOSLYN is the President of SJ+ Systems Associates, Inc. and is the author of PRC?, a complete, integrated software development life-cycle management / IT Governance tool for U2. She has worked with U2 (nee Pick/Multivalue) and SB+ software the beginning (both hers and its) and has specialized in IT Governance, including quality, compliance and life-cycle productivity issues since the early 1990's.

Prediction 2012

BY CHARLES BAROUCH

Every year brings new challenges. I'm not usually fond of predictions. Disruptive events can and do occur. Still, we have to start from some assumptions if we want to have any chance of being prepared for what is coming. So, with my crystal ball charged and ready, I give you my best guesses. You can feel free to supply your own grain of salt.

Smart Phones

This will be the year of smart fatigue. We've had our expectations revised so many times that we cease to be amazed. We expect smarter, smaller, smarter, faster, and smarter. The companies making these devices will keep delivering that. Other than fanboys, most people aren't going to care about any given new feature. Note: I am something of a fanboy.

My prediction: Phone news will start getting overshadowed by tablet news. Phones will still sell better than tablets. We will see a lot of cool stuff. Flexible screens are looking likely, for example, but none of it will be as game-changing as the breakthroughs we have already seen. We've seen this in the spreadsheet world — new features every year,

We still live in a world where the smart developer builds pretty and functional together. If you know how to make things work, you'd better start learning how to make it look right.

but each set gets more esoteric and less generally useful. We've had the wow for now. It may heat up again in 2013, hard to say. If you need a smart phone strategy, odds are you really need a tablet strategy that scales back down to smart phones.

Tablets

Tablets are finally coming of age. This is the year Microsoft will try to regain their mojo in all things beyond the desktop. The year Apple tries to retake market leadership in phones while preserving tablet dominance. The year Google polishes up Android while competing with alternate Android forks from Amazon, Baidu, and others. HTML 5 will be positioned as an alternative to choosing sides.

My predictions: Apple's share will continue to slide, well into 2013. When the slide halts, they will still have more than 50% of the market. People mistake the re-balancing of the playing field as bad news for Apple. They are wrong. The correction was inevitable. Apple, however, got the opportunity to define the field and for many people, anything not Apple has more to prove. If you design for tablet, you have to consider first HTML5, then Apple, then Android, then anything else. Microsoft will not do well because they have had this market for longer than Apple and still haven't figured out how to play in it. They won't get smarter fast enough for 2012.

Cloud

This is going to mean more to consumers than to business. Apple has assured that by getting the consumer conversation started. We in the business community already know about the cloud. It isn't really new to us, we've seen timeshare. We know what central data repositories and central application services do for us and for business. I believe 2013 will be a better year for the professional cloud. It will take until then for the idea to get properly marketed and packaged with the right offerings.

My prediction: There will be some winners in the corporate space in 2012. They are unlikely to be new; they will be 2011 or earlier players repositioning. Smaller winners in cloud will be the rule. These people will get no headlines, they will however, make money in their various niches.

Databases

SQL has become a dartboard for media. The No SQL movement is going to continue to make the news.

My prediction: There is still going to be a lot of SQL. Don't count it out yet. Microsoft, IBM, Oracle, and others will still push a lot of product out the door. It will take time for businesses to understand why they should change. Look at Semantic and other, search-adjacent, databases to make more news and possibly make some actual progress. In the non-SQL world, as in the past, Applications will make the difference. The cloud could be the MultiValue world's best friend, except Cloud is not going to win big in business this year. We will slog through; we will get small-to-medium wins. Unless some application drags us through to a big enough spotlight, we will continue without a spectacular win in 2012. We will continue.

Applications — Other than Phone/Tablet

Custom programming and programming staff will become more important as people continue to try to build software which cuts down the head count while maintaining speed of execution. On the Product side, expect more and more tablet and phone apps. Desktop/laptop apps will still make money, but it will be much harder to get new ones to market in any big way. As we have seen in the past, web delivery to a browser is going to beat out GUI in most cases. I still make money with Delphi. Increasingly, I use it for the backend and present the interface via the web.

My prediction: Usefulness will still trump method of delivery in many applications, but slick interfaces will always sell a lot of crap. We still live in a world where the smart developer builds pretty and functional together. If you know how to make things work,

Continues on page 19

jQuery Mobile

Making Mobile with Minimal Misery

BY KEVIN KING

In the summer of 2010, a group of highly ambitious and talented developers started working on a project to create an elegant HTML5-based mobile application development platform. Building upon the popular jQuery Javascript library, jQuery Mobile was designed to support an amazingly long (and growing!) list of tablets, mobile phones, e-readers, and even desktop systems.

This past November, following an extensive alpha and beta testing period, jQuery Mobile 1.0 was released. Even more astounding than the list of supported devices is how easy this platform is to use! Anyone with even a smattering of HTML experience can now write code for a wide variety of mobile devices. Furthermore, while the end result is a web application — considered by some to be inferior to a full-fledged mobile application — with up-and-coming bridging technologies like PhoneGap (www.phonegap.com), web applications can now reach beyond the browser to access the various hardware components available in popular phones and tablets.

If you're wondering why mobile development is so important, consider this — People don't carry around Wyse 50's in the hip pocket.

If you're wondering why mobile development is so important, consider this — People don't carry around Wyse 50's in the hip pocket. They do, however, carry around cell phones with browsers. If you have a product or service to sell, wouldn't it be nice to be able to reach out to the millions of potential customers who use these devices every day?

Prior to jQuery Mobile, coding for mobile devices could be quite challenging. Different devices required different approaches, tools, and methodologies for application construction. And writing an app for the iPhone was a completely different process than writing an app for an Android device. This has been complicated even further by the dozens of new mobile devices being thrust upon the market each year.

With jQuery Mobile, things are so much simpler! Using advanced web development techniques like progressive enhancement, CSS media queries, and semantic HTML (things you don't need to know about unless you really want to know more) jQuery Mobile allows you to write simple HTML pages with built-in interactions without having to fuss with complicated Javascript or CSS. (Actually, while jQuery Mobile is built upon the popular jQuery Javascript framework, you don't have to touch ANY Javascript or CSS unless you really want something unique.)

Figure 1 shows the source code for a simple menu rendered in jQuery Mobile. Figure 2 shows what this looks like in the mobile browser. But rather than typing this in, you can simply view the documentation at intl-spectrum.com/s1046 and cut and paste most of this code.

Note the line that reads:

```
<ul data-role="listview">
```

We can add much more functionality to this menu by adding a couple of additional options to this tag. Note that we're not adding any functional code, just options. For example, let's change this to:

```
<!DOCTYPE html>
<html>
    <head>
    <title>Page Title</title>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
</head>

<body>
    <div data-role="header">
        <h1>My Mobile App</h1>
    </div>

    <div data-role="content">
        <ul data-role="listview">
            <li><a href="welcome.html">Welcome!</a></li>
            <li><a href="products.html">Our Products</a></li>
            <li><a href="services.html">Our Services</a></li>
            <li><a href="about.html">About Us</a></li>
            <li><a href="contact.html">Contact Us</a></li>
        </ul>
    </div>
</body>
</html>
```

Fig.1

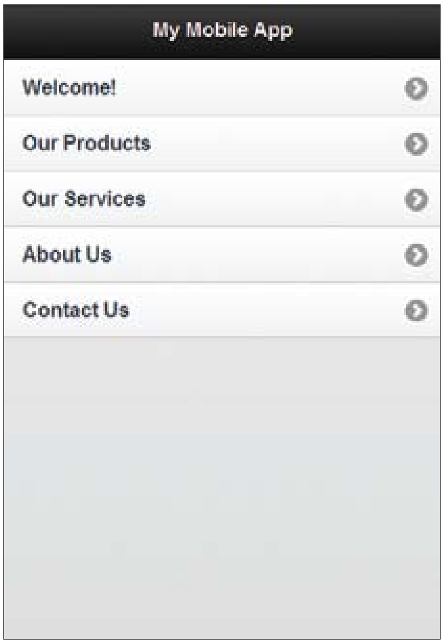


Fig. 2

```
<ul data-role="listview"
  → data-inset="inset"
  → data-filter="true">
```

See Figure 3 to see how this renders on the mobile device. The "data-inset" option sets the menu in a little bit from the sides as an accent, and the "data-filter" option adds a search option on top of the menu. Figure 4 illustrates how this search feature works; type in anything displayed in the menu and the menu will filter itself to show you only the options that include the letters you've typed.

This is only one of many cool things that jQuery Mobile offers. If you have a spare moment, use your phone, tablet, netbook, e-reader, or even a full fledged computer to browse over to www.jquerymobile.com to see all that this platform can do. And if you want to know more, come to my sessions at Spectrum 2012 where I'll be demonstrating, discussing, and diving much deeper into this amazing new tool! IS

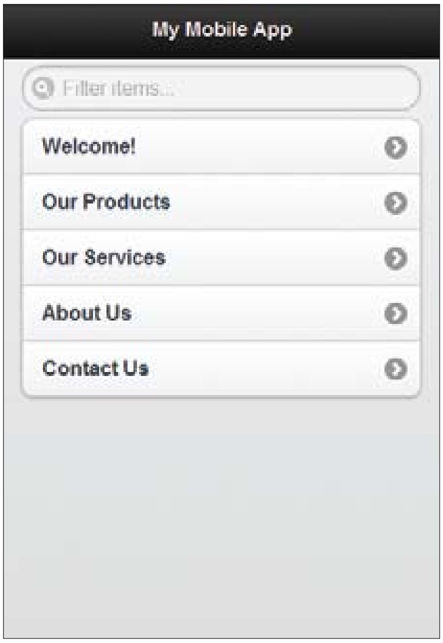


Fig. 3

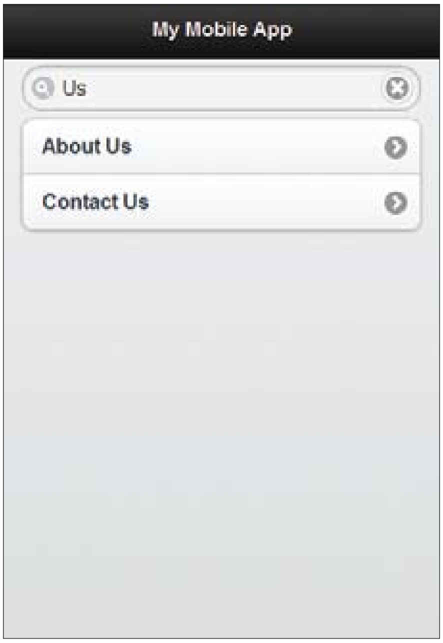


Fig. 4

U2 Replication

An Overview of our Scalable and Robust High Availability Solution

BY JONATHAN SMITH
LEAD TECHNICAL SUPPORT ENGINEER, ROCKET SOFTWARE

Ten years ago the necessity for a true 24x7x365 environment was regarded as the exclusive right of high-end enterprise-level systems and applications. Since that time, the industry and customer requirements have moved forward, and we have all come to rely on such systems in our day-to-day lives. Consequently many of our customers' applications developed on Rocket Software's UniVerse are now required to deliver this level of availability. Prior to the release of UniVerse v11.1 this would have been achieved using a combination of hardware-assisted replication and the Transaction Logging capabilities of UniVerse.

With UniVerse v11.1, Rocket has dramatically increased UniVerse's ability to provide a highly-scalable, robust 24x7x365 environment.

How Is This Achieved in UniVerse v11.1?

11.1 includes U2 Replication, which is based upon the proven field-tested UniData Replication. The transaction logging capabilities of UniVerse can be used to complement U2 Replication,

U2 Replication's primary use is as a mechanism to provide a robust, scalable Disaster Recovery (DR) / High Availability solution, and can also be used for offloading an inquiry workload.

but is not a mandatory requirement for deployment.

What is U2 Replication?

U2 Replication provides an automated, scalable, robust method of delivering read-only copies of UniVerse files and their transactions from one data server to other servers, and also allows the propagation of various account level operations such as compilation and cataloguing of programs, and the creation and deletion of files between the data servers.

Available on Windows and UNIX platforms, it can operate in Real-Time, Immediate, or Deferred modes. U2 Replication also gives users the added ability to set up a standby / failover and failback relationship between servers and supports "Cross Platform" Replication, meaning that publishing and subscribing systems do not need to be on the same platform.

U2 Replication supports both account-level and file-level replication. Account-level replication is a configuration style that allows you to simply replicate all data files defined in the VOC file of a UniVerse account, and file-level replication is a configuration style where the data files to be replicated in the VOC file of a UniVerse account are individually declared.

Where Can We Use U2 Replication?

U2 Replication's primary use is as a mechanism to provide a robust, scalable Disaster Recovery (DR) / High Availability solution, and can also be used for offloading an inquiry workload. For example, it gives users the ability to have a dedicated reporting system taking the potential resource-heavy user reports away from the production system, allowing more resources for time-critical response screens.

U2 Replication can also be used for distributing data to regional centers for local processing. For example, the user can move data to regional centers for processing by that center, and you could also have the regional center publish its results back to the central server as well.

What Are Some of the Concepts Involved?

A system on which data is maintained and distributed to other systems is referred to as the **publisher**, and a system that receives data from a publisher is referred to as the **subscriber**.

Multiple subscribers can be associated with a single publisher and vice versa. U2 Replication is publisher driven; that is, the publisher pushes transactions to the subscriber.

A **Replication Group** is based on an account and is used to define the configuration style of that account. Each group is defined as an account-level or file-level replication and has its own set of tuneable resources. These include:

- The number of writing processes on the subscriber
- The size of the memory structures used
- The publishing and subscribing systems
- The replication mode
- The action to be taken if an error is encountered

An account can have one account-level group defined, but it may also have several file-level groups defined for it, as well. This allows for a highly scalable and tuneable configuration in terms of performance.

Account-level replication is an inclusive configuration that allows the exclusion of files. U2 Replication provides two methods of declaring files which you do not either need or want to be replicated. These can be declared on a group basis or as a general declaration to be excluded from all groups.

By default, the files on a subscribing system are read-only. This could become problematic if the subscribing server was to be used as a reporting server as, when logging in, applications may attempt to update their own logging files. In order to allow for this, files can be marked as writable on the subscriber by exception. Any updates on these files on the subscriber are not sent back to the publisher.

U2 Replication introduces new processes, memory structures and associated architectural changes in UniVerse v11.1. An overview of the U2 Replication architecture is shown in figure 1, which explains how a single replication group running normally is processed in UniVerse 11.1.

The yellow boxes represent physical processes, the boxes in dark blue are memory structures, the files in light blue are physical files, and the boxes in salmon are a tool or command line interface.

How Do I Monitor U2 Replication?

Rocket is aware that many of our users will look to deploy U2 Replication with their application into their customer sites where there is little or no UniVerse DBA presence. In order to support this situation, there are several controls built into U2 Replication to aid deployment and monitoring.

U2 Replication has a "heartbeat" and an "exception action script" mechanism. When the system is idle, a "heartbeat" packet is sent approximately every four seconds between the publisher and the subscriber. If the "heartbeat"' goes unacknowledged, then replication is suspended and the 'exception action script' is sent. The "exception action script" could be used to email notification of the suspension and take a recovery action automatically. Each replication group is also configurable to take advantage of this script when an error is encountered.

To allow for an easy visual check on the status of U2 Replication there is a GUI Replication Monitor (fig. 2), which is based on a traffic-light system in order to make it very easy to understand.

During normal operation the 'Sync Status' moves from green to yellow and

Continues on page 18

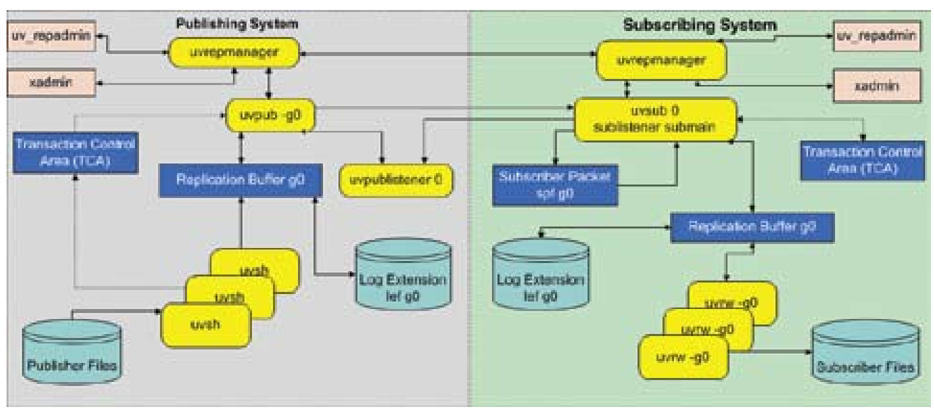


Fig. 1

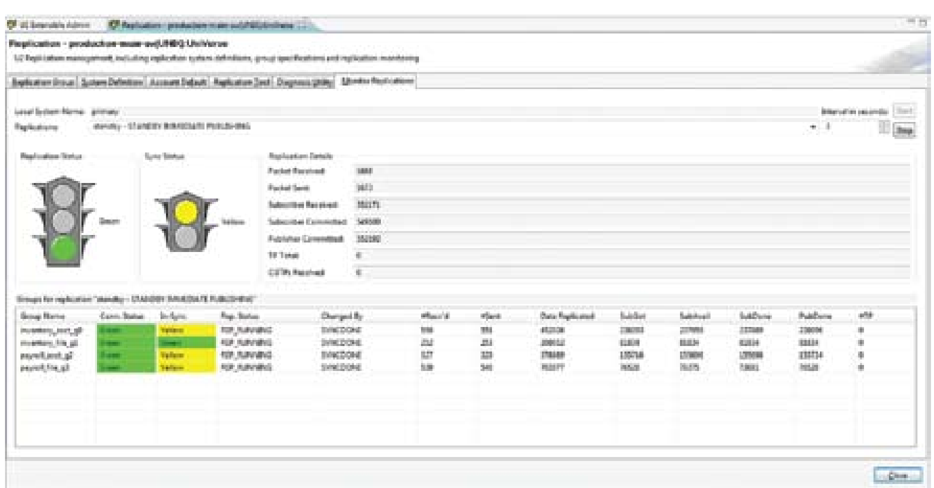


Fig. 2

From the Press Room

Datatel and SunGard Higher Education Combination Clears Regulatory Review

Hellman & Friedman, Datatel, SunGard Data Systems and SunGard Higher Education today announced that the proposed combination of Datatel and SunGard Higher Education has cleared Department of Justice review and the companies are preparing to close the transaction in the first quarter of 2012. As a result of this combination, over 3,000 employees will be dedicated exclusively to delivering solutions to education institutions in 40 countries.

"We have heard from many customers that they are eager to take advantage of the benefits that will be provided by this combination," said John Speer, Datatel President and CEO. "We expect to soon be in a position to deliver on this promise."

As noted when the proposed combination was announced on August 5, 2011, customers of the combined companies will benefit from:

- Continued investments in current solutions: The combined companies plan to support the solutions of both companies going forward and to continue to invest significantly to enhance those solutions.
- A broader portfolio of solutions: Customers will enjoy powerful new capabilities from the combined portfolio of products and services offered by Datatel and SunGard Higher Education.
- Accelerated innovation: The combined companies will pool resources and talent to accelerate research and development with targeted investments focused on major challenges in the education community.
- An expanded knowledge-sharing community: Institutions will experience increased collaboration, creativity and knowledge-sharing across an expanded global customer community of over 2,300 colleges, universities, foundations and state systems.

"This combination will help us achieve our goal of better serving customers and education communities globally," said Ron Lang, SunGard Higher Education CEO. "Our largely complementary products and services portfolios blend well, and our deep, shared commitment to our customers will continue to fuel the long-term relationships and innovative spirit we value so highly."

As described in the initial announcement of the proposed combination, affiliates of private equity firm Hellman & Friedman LLC will acquire the SunGard Higher Education businesses from SunGard Data Systems Inc. and combine the acquired businesses under a new holding company with Datatel, an existing Hellman & Friedman portfolio company. Closing of the proposed transactions continues to be subject to the satisfaction of customary closing conditions. ■

Entrinsik Announces Release of Informer Web Reporting With Enhanced SQL Functionality

Entrinsik Inc. (www.entrinsik.com), developer of award-winning web-based reporting and analysis solutions for organizations across a variety of industries, recently announced the availability of powerful new SQL functionality for Entrinsik's Informer software. The new Informer Native SQL reporting brings an unprecedented level of scalability and flexibility to organizations using Informer against SQL databases.

Informer software provides real-time access to multiple databases including Oracle, SQL Server, MySQL, Access, DB2, Informix , U2 and others without complex ETL processes, data warehousing, or data cubing. From an intuitive drag and drop web interface, self-service reporting capabilities enable end users to drill down into data elements in real-time, sort and group, build interactive charts and perform ad-hoc analysis without IT support. Informer is implemented in hours and is easy to install, configure and administer, delivering quick ROI and a low total cost of ownership.

With the new functionality, Informer users can now enjoy the flexibility of creating an Informer report directly from a SQL query or stored procedure. Users who have existing SQL query statements now have the ability to immediately reuse these queries to create an Informer Report without the need to set up the metadata and mappings. Implementation and roll out is now faster than ever.

"Informer's new capabilities reflect our ongoing commitment to help organizations analyze data in real time, regardless of where it's stored, in order to make the best decisions possible," said Doug Leupen, President/CEO at Entrinsik. "Our developers continue to raise the bar with innovation that enables us to deliver a powerful yet easy-to-use reporting platform."

Watch a recorded webinar demonstrating the new Informer SQL functionality at http://bit.ly/InformerSQL.

Over 1,000 organizations around the world use Entrinsik's software every day. For a personalized demonstration or a free trial, call us today at 888-703-0016 or visit www.entrinsik.com.

Entrinsik, Inc.

Based in Raleigh, NC since 1984, Entrinsik develops, implements, and supports Information Management Solutions that enable organizations to maximize performance and improve bottom lines. Entrinsik's Informer

Web Reporting is an innovative, award-winning reporting and analysis solution used by tens of thousands around the world. For a free trial, visit www.Entrinsik.com. or call 888-703-0016. ■

Revelation Software announces the release of OpenInsight Development Suite 9.3

Revelation Software announces the availability of the latest release of OpenInsight Development Suite (OI) 9.3. OI 9.3 provides the following new components:

- Data Encryption at Rest (RTIDER) – OpenInsight version 9.3 incorporates a new encryption service that system administrators can deploy to encrypt (on a field by field level) data in OpenInsight. The data will be "encrypted at rest" using industry standard encryption routines. Clients and the encryption service use the Windows Communication Foundation (WCF) to communicate, thus allowing the data to be encrypted in transit. Through the use of an Encryption Server (normally, but not necessarily, the same machine as the Universal Driver server), each OpenInsight workstation seamlessly and securely encrypts data for storage when written into an encrypted volume.
- OpenInsight for Web (O4W v1.2) – This version of O4W includes many enhancements to the API's, Form, Report, Dashboard and Menu wizards. O4W v1.2 also provides additional support for mobile devices, such as tablets and smartphones. Support for these mobile devices have been implemented as a series of additional API's that can be used in building O4W stored procedures. Using jQuery Mobile, the companion library to O4W's jQuery, web pages can be built that follow the design styling that is now becoming the standard on smaller devices.
- Banded Report Writer – The original Banded Report Writer (BRW) has been replaced using technology based on the C1Report tool from ComponentOne. The OI BRW supports powerful and easy-to-use banded reports that are compatible with Access and Crystal models.
- Extended math functions – OI 9.3 includes new extended math functions: _addx, _subx, _mulx and _divx. These functions allow developers to perform math operations using full precision or a specified precision.
- OI for D3/mvBase Connector – The OI for D3 connector provides the ability to connect with both the D3 and mvBase databases from TigerLogic Corporation.

"Our connection technology has truly made OpenInsight a versatile front-end graphical database development environment", said Mike Ruane, President/CEO Revelation Software, "Whether your backend database is Linear Hash, U2, D3, mvBase, OpenQM, NoSQL or SQL our tool set leverages MultiValue skills with modern technology. Also included in the OI 9.3 release are enhancements to the following components: System Editor++, Database Manager, Basic+, O4W, OIPI.net/OIPI, Arev32, Table Builder, Report Builder, IDE, RDK, U2 Connector, SQL Connector, and Module Management.

"Data encryption and mobile computing are fast becoming essential requirements with any application development," said Robert Catalano, Director of Sales, Revelation Software. "We are empowering our development community with one of the best mobile computing and data encryption toolkits on the market."

OpenInsight Development Suite 9.3 is available from Revelation Software, or through their network of resellers. See Revelation.com for details.

About Revelation Software

Founded in 1982, Revelation Software delivers a suite of application development tools and companion services that take full advantage of leading network computing architectures, messaging, and operating environments. Today, the company's flagship product OpenInsight is the only database development suite that provides both Windows, Web 2.0 and .Net tools to develop and deploy mission critical applications. There are more than 1.5 million licensed users of Revelation products across 60,000 deployed sites worldwide. The company has offices in Westwood, New Jersey, as well as a European distributor in the United Kingdom, and an Asia Pacific subsidiary in Australia. ■

System Builder 5.4 and below go to End of Service

As of December 31, 2011 both SB Runtime (SB+), versions 5.4.x and below and SB Client (SBC), versions 5.4.x and below will be moved to End of Service and End of Life.

Effectively this means

- Technical Support will no longer maintain these versions for problem reproduction.
- The product version has been withdrawn from sales.
- New or additional licenses are no longer available.
- Replacement media is not available.
- Maintenance IS available.

Customers on maintenance MAY call for assistance, access the on-line knowledge base and obtain product upgrades. For further information, contact Rocket Software at u2askus@rocketsoftware.com. ■

U2 REPLICATION: AN OVERVIEW OF OUR SCALABLE AND ROBUST HIGH AVAILABILITY SOLUTION

Continued from page 15

the 'Replication Status' should remain green. It will move to yellow as a result of a DBA-initiated activity, such as suspending a replication group, and move to the red in the event of an error. This allows for an easy visual check of the overall status.

What Else Has Rocket Provided to Aid and Complement 24x7x365 in UniVerse v11.1?

UniVerse v11.1 now provides the ability to a rebuild an index while the file is in use, adding to the ability to a resize a file while it is in use. This eliminates lengthy downtime for these routine file maintenance procedures.

Along with U2 Replication Rocket also introduced External Database Access (EDA) to UniVerse v11.1. EDA can be used in conjunction with U2 Replication to transform your replicated data and consolidate it into an external data source such as SQL Server, DB2 or Oracle for use in reporting and business intelligence.

Summary

With the introduction of U2 Replication and complimenting features UniVerse v11.1 now offers a scalable, robust, feature-rich environment for a 24x7x365 operation for any size of business, with the addition of real-time interoperability with other databases. IS

JONATHAN SMITH has worked with U2 for 25 years covering many roles. His current role in Rocket is as a Technical Lead for both data servers, consultant and worldwide trouble-shooter.

LETTER TO THE EDITOR

Kevin King's modern complaint about our legacy connections (serial or telnet) missed one incredibly major, major, major, most important, can't think of anything else more limiting reason why persistent connections are absolutely necessary.

It's the INPUT statement. Period.

I don't care if you have a 500 GHZ screaming server with 10 terabytes of memory and 100 petabytes of storage. If you have a Pick application that the users's session comes upon the INPUT statement, all of that horsepower comes screeching to a halt while the user contemplates entering their next answer.

It's that simple. The INPUT statement has been the hardest to 'one-size fits all' command replacement for Pick systems.

Enough complaining. Should I write an answer article espousing the corner that legacy (perhaps all) standard Pick software has painted ourselves into with the INPUT statement.

As long as there are green screens (yes, another sore topic but not to be overlooked), the INPUT statement prevails.

I could write a book (an article) on the history of replacing this single most system-dependent command (along with PRINT) and some half-baked 4GL ideas or subroutine that attempted to work around it (some good, some bad).

But as long as the net result is either INPUT or IN, then the zillions of legacy programs (not to mention procs) are still going to be dependent on this Persistent Connection.

Mark Johnson

Kevin replies: *While you make a good point about the INPUT statement and how it can be problematic for converting existing applications, it would appear that you and I are looking at this from different perspectives. In our MultiValue systems, whether a record in a file was updated from a green screen, GUI, external connector, web front end, whatever, the most important result is that a record in a file gets updated. We shouldn't be focusing on trying to wrap some new front end around our existing screens; rather, the better investment is to capture the input from whatever device offers it, and then pass that information into INPUT-less routines that will perform the updates regardless of the source of the input.*

Without question there are numerous programs in our legacy applications that suffer from a seemingly inextricable commingling of input, processing, and output in the same routine. Validation routines with multiple messages and exit points can be especially problematic. While this type of programming will inevitably will be more difficult to convert to a new input methodology, the task is by no means impossible. Furthermore, while some companies may choose not to refactor their existing applications, that shouldn't stop us from considering these new technologies and methodologies in our new development.

BUSINESS TECH: PREDICTION 2012

Continued from page 11

you'd better start learning how to make it look right.

Operating Systems

Mac, Windows, Android, iOS, Linux, Unix… we have an embarrassment of riches. Personally, I like a world with lots of choices. One size has never fit all in my view. Desktops will keep declining but they still represent a huge percent of the landscape. As the year goes on, they will keep losing importance.

My prediction: At the end of the year, despite Cloud data, data in your pocket (phones), and data in your briefcase (tablets, laptops), most business will still involve servers. Those servers will be predominantly running Linux, Unix, and Windows. This will change less slowly than the headlines suggest.

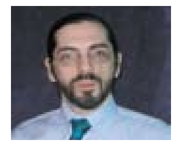
The Economy

2011 was not a banner year for any major economy, anywhere in the world. The challenges have left a lot of talented people without work. The outlook has been bleak for the unemployed and nervous for those with jobs. This isn't an IT thing, it's an everyone thing. People still haven't come to grips with the idea that many of the jobs that disappeared are not overseas or anywhere else where they might "come back" from any time soon. Until we see stronger anti-monopoly laws, jobs made excess through mega-mergers aren't coming back. Other jobs are gone because technology and other efficiencies have permanently eliminated the need.

My prediction: 2012 will be better, but we won't be seeing a second coming of the 1980s or any other boom time. It will be better in part because our definition of better has been beaten into submission. Expect IT jobs to start opening up more, but salaries will be all over the charts. The economy will slow adoption of every new tech because they all require money and time.

Place Your Bets

Whether you think you are betting or not, we all do. Every day we accept or reject seminars, webinars, magazine articles… all sorts of opportunities to expand our domain. A lot of us can get by betting on doing nothing, but that is shrinking job market. IS



CHARLES BAROUCH is the CTO of HDWP, Inc. You can see him in person at the International Spectrum Conference in Florida.

Rolling Your Own

MultiValue

Web Connector

Part 1

BY KEVIN KING

If you've been following these articles for the past few months, you may be sensing that I'm a bit stoked about all that's happening in the world of web technology. And why not? With new versions of HTML, CSS, and Javascript fresh out of the oven, plus an abundance of new web-enabled tablets and phones hitting the market, there is truly MUCH to be excited about!

All the cool web technology available right now doesn't mean much if we can't make it work with our MultiValue systems, however. Our information, the valuable lifeblood of our companies, lives in those systems, so if the web can't talk to our databases then … why bother, right? Well my friends, prepare yourself to bother because I'm about to show you that a browser CAN talk to MultiValue databases, and it's not nearly as difficult or expensive as you might have been led to believe.

Digital Version On-Line

Scan Me!

A browser CAN talk to MultiValue databases, and it's not nearly as difficult or expensive as you might have been led to believe.

Before we get too far down this rabbit hole, allow me to first acknowledge that there are a number of vendors who have spent a good deal of time, money, and effort creating solutions to make the web accessible to MultiValue programmers. My point here is not to discount those efforts or to undermine them in any way. Rather, for those who don't mind a bit of bare metal programming and diving deep into the dirty bits, I'd like offer you something interesting to think about, something fun to "research," and if all goes to plan, this exercise might just *shift* your perception a little.

In an effort to keep this simple, we'll be building this connector as a series of incremental steps. Today we'll create the MultiValue parts needed for our connector. Later we'll create some code to connect our MultiValue solution to a web server. Finally, we'll tie it all up with one more script that will give us secure and controlled access to our MultiValue data and programming from a typical website.

Ready to jump into the dirty bits? Let's do it!

Most MultiValue systems have a shell program that is launched from the host operating system command line. For example, Unidata has "udt", QM has "qm", Universe has "uv", etc. While these shell programs are usually launched from a login script when someone makes a telnet connection, these commands can just as easily be launched from an ssh session, cron job, or even from a web server. This is this foundation on which we will build our connector.

On the MultiValue side, our connector is nothing more than a small Basic program that accepts some input and does something practical. Of course, "something practical" could mean just

Continues on page 22

UniVerse® V11.1

Add powerful high availability, interoperability, administration, and security capabilities to the solid extended relational database.

UniVerse®

The Reliable and Flexible Foundation for Business Solutions

Rocket | U2

- U2 DataVu • U2 Web DE • U2.NET
- IBM.NET • SB/XA • wIntegrate®

For more information, email: U2AskUs@rocketsoftware.com
Visit: http://www.rocketsoftware.com/u2/products/universe

U2 Migration Factory

Migrate to the Best MultiValue Databases Available: UniData® or UniVerse® – for Solutions Powered by Rocket U2

Ask about our free migration offer.

Rocket®

ROLLING YOUR OWN MULTIVALUE WEB CONNECTOR – PART 1

Continued from page 20

about anything, so we'll build our routine as a simple wrapper that can call a subroutine. With this foundation in place, your team of geniuses can create subroutines for whatever functionality your web solution may require.

Before we create this program, however, we need to figure out how to ask our connector to do something. Most MultiValue systems provide some way of reading environment variables from the underlying operating system, so let's start there. We'll set an environment variable called WEBREQ with a request, start up our MultiValue database, and then grab the response when the shell finishes.

Another thing to consider is how we will launch our Basic routine. Most systems have some way of starting the shell and launching a specific program, but they don't all work the same way. Therefore for portability we'll leverage the login procedure (named "LOGIN" on Unidata or may be named according to the account on other platforms) to check for our environment variable and do something accordingly. If our environment variable is not set (as would be the case for a normal telnet connection) our program can stop and allow the login procedure to continue normally.

Figure 1 shows how this might be accomplished using a program named RUN.WEB.REQUEST. This program was written for Unidata but could be easily translated to a number of MultiValue platforms.

In this program we read the environment variable WEBREQ and expect it to contain three comma-delimited pieces: 1) the subroutine to be called, 2) the name of a file where the output will be written, and 3) the input data for the routine. Once we've parsed the three parameters from the environment variable, we use a CALL @ to call the subroutine as named and then write whatever result was returned to a directory file named WEBOUT (which I usually map to the _HOLD_ directory on Unidata or $HOLD on QM).

Despite the simplicity of this routine, there are many details to explore. For example, you might wonder why we're passing the input in an environment variable but writing the output to disk. Good question!

The input from the web might contain confidential information (i.e. credit card numbers) so we don't want that information EVER written to disk. Of course, with the output written to disk — the only portable way to get information out of the database and into the underlying operating system where it can be picked up by an external program — we must be careful to avoid including sensitive information in our output or find another way to get the information out of the database shell and into the calling operating system.

(Printing the output to the screen and capturing it in a calling script can work too. However, I like to use CRTs and COMO for debugging and having debugging output mixed with the real output data is a little too inflexible for my tastes. Also note that this routine leaves basic error handling as an exercise for the reader.)

Rather than only calling subroutines, we could expand upon this concept to include fundamental database operations like reading, writing, deleting, returning a select list, executing a TCL command, etc. However, because these operations necessitate more advanced security considerations, we

```
*
* Program Name: RUN.WEB.REQUEST
* Written By:    Kevin King
* Project:       Spectrum Demo
* Date:          26 Nov 2011
* Description:  This routine will look for an
*               environment variable WEBREQ and
*               will call a subroutine as requested
*               when found.
*
WEBREQ = GETENV('WEBREQ')
IF (WEBREQ NE '') THEN
   OPEN 'WEBOUT' TO F.WEBOUT THEN
      SUBR.NAME = FIELD(WEBREQ,',',1)
      OUT.NAME  = FIELD(WEBREQ,',',2)
      DATA.IN   = WEBREQ[COL2() + 1,999999]
      DATA.OUT  = ''
      CALL @SUBR.NAME(DATA.IN,DATA.OUT)
      WRITE DATA.OUT ON F.WEBOUT,OUT.NAME
      CHAIN 'OFF'
   END
END
*
STOP
END
```

Fig. 1

```
SUBROUTINE TEST.SUB(DATA.IN,DATA.OUT)
*
DATA.OUT = OCONV(DATA.IN,'MCL')
*
RETURN
```

Fig. 2

will be focusing only on calling subroutines. These subroutines can then be written as securely or insecurely as you prefer.

Speaking of security, you've probably also noticed that our routine allows a subroutine to be called with absolutely zero authentication or additional security. Many MultiValue systems don't ask for additional authentication once someone has logged in to the box, which may be fine for telnet users, but should be given additional consideration for web-based connections.

Figure 2 is a very brief example of a subroutine that might be called from our connector. Note that this subroutine takes two formal arguments, one for input and the other for output. In this simple example we're converting the input string to lower case, but as was mentioned earlier, this subroutine could do absolutely anything it wants with the input in order to produce the output.

With nothing more than these two routines and a minuscule edit to the login procedure to call our starting program, we now have a way of invoking a MultiValue subroutine from the outside world. Next time we'll take a look at how we can use the open source Apache web server and a remarkably small bit of PHP to provide this functionality to a larger audience! IS

KEVIN KING is the President and Chief Technologist with Precision Solutions, Inc., a leader in technology solutions, support, and training. He is also the author of SB+ Solutions, an enthusiastic private pilot, and Christian guitarist and producer... as time allows.

BY SHANNON STOLTZ

FINDING THE Value *of* Training

Training is an investment. Not a tangible investment like information technology, but nonetheless, an investment. It is an investment in the future of your company, in your department, and in your employees. The trick is finding the value of the training investment in a tangible, quantifiable manner.

What is training?

First, what is your perception of training? What is training? For some, "training" brings up government regulations for discrimination and cultural diversity training. For others, the word training conjures up negativity like "expensive," "boring," and "unproductive." Some of us think of training with glee, an opportunity to grow and learn. Simply speaking, training is any opportunity to improve skills and knowledge, an opportunity to learn. This magazine, International Spectrum, is a training tool in itself. Albeit a free one, but education, learning, and skill building can come from reading its pages. Training also comes in the form of informal gatherings of programmers sharing knowledge and ideas, brown bag lunch sessions, user group meetings, webinars such as those offered from International Spectrum or others, customized on-site training from database and tools vendors, off-site training or "boot camps" from your database providers, and conferences and trade shows like the International Spectrum conference or your vertical market conferences. Even mentoring programs, either internally-based or from vendors, provide a training component. For the IT professional, training encompasses more than just technical topics, but also interpersonal communication, leadership, business analysis, project management, customer service, and even conflict management. Why? Because training improves the bottom line — if it is provided in a focused, results-oriented manner.

Even mentoring programs, either internally-based or from vendors, provide a training component

Digital Version On-Line

Scan Me!

Focusing your Training Efforts

Training costs can add up. So how do you ensure you are providing the right training for your employees (or yourself)? By providing training that increases the business' ability to grow, adapt, and change in its marketplace and grows the employee's skills in a manner that enables and equips them to perform well, you can increase the effectiveness of your training dollars.

Choose training opportunities that allow your employees (or yourself) to grow in knowledge and skills that further the business. Look for opportunities to increase productivity, lower technology related costs, and be on the lookout for tools that will improve efficiency and/or quality. If you are in the software business, programmers need to understand the importance of quality and testing; programming bugs (or undesirable features) cost time and money in support, programming patches, and customer service relations.

Another approach is to build employee skills based on performance and personal development needs and interests. Employees who are engaged in the training process and are growing also tend to be engaged in the business process and look for ways to enhance the business. By growing your employee skills, you create a more flexible, adaptable workforce to meet business needs and a more satisfied workforce.

Identifying the Value

When looking at training's potential value or measuring its effectiveness, ask yourself, "Will the training…

- be used in the workplace?"
- reduce costs?"
- improve productivity?"
- improve business processes?"
- improve system, network, application, or database performance?"
- increase the quality of our service, application, or business?"
- reduce wasted assets, time, or opportunities?"
- expand our ability to produce new products or services?"
- give us more flexibility to adapt, produce, or develop?"
- increase employee or team productivity?"
- allow us to maximize the use of a product or service we have now or will have in the near future?"
- improve our internal or external customer satisfaction?"
- improve our employee satisfaction?"
- cut our cycle time or improve our schedule performance?"
- be useful to others in our organization?"

Factor in the value of your answers and compare it to the cost of the training. One training session may provide the insight or knowledge needed to save hours of labor each week. That cost savings may pay for the training.

Maximizing your Training Investment

One way to cut the cost of training is to leverage it. Teleseminars, webinars, and online training seminars usually allow you to pay one fee for the entire site, allowing the training cost to be split across as many employees as you can fit in the same room in front of the computer and a speaker phone. The

Continues on page 26

FINDING THE VALUE OF TRAINING

Continued from page 25

same leveraging concept can be applied other training venues as well.

If a training session would be helpful to more than one employee, it may be more cost-effective to arrange an on-site training session, bringing the trainer to you. For employees who participate in off-site training sessions, including conferences and trade shows, arrange for the employee to prepare a series of short (one hour or less) sessions with their co-workers to share the knowledge. Arrange for optional brown-bag training sessions at lunch times, where an employee shares a skill, or tips and tricks with others. This technique not only spreads the knowledge, but also builds your employee's presentation and communication skills as well.

Encourage your senior staff members to share their knowledge with others, creating mentoring opportunities and enhancing your employees' skills. If you have a project but don't have the skill set on hand to do it, consider creating a mentoring relationships between consultants and your staff, so the project gets done and your staff gain new skills for future projects.

The true value of training investment comes when the knowledge and skills compound and grow among your staff. After all, a company's intellectual capital is often what gives it the competitive advantage it needs to succeed. IS


INTERNATIONAL
Spectrum

Tech Tip

Sort Data by Week Number

There are times when doing reports when you need to sort your values into weeks. Most people would end up writing a program to do this, but that is not necessary.

So assuming attribute 14 contains a date, then

```
F Correlative
F;14;C7;/
A Correlative
A;(14 / “7”)
```

A full dictionary item would like the like the following

```
WEEK.NBR
001 S
002 0
003 Week Number
008 F;14;C1;+;C7;/
009 R
010 6
```

is now the week number since 1968, with Sunday as the start of the week. You can then use this in a BREAK-ON.

```
SORT FILE BY DATE BREAK-ON WEEK.NBR
```

If you want to have Monday as the beginning of the week, then do the follow:

```
WEEK.NBR.M-S
001 S
002 0
003 Week Number
008 F;14;C1;-;C7;/
009 R
010 6
```

If you want to sort based on the week number within the year, use:

```
WEEK.NBR.IN-YR
001 S
002 0
003 Week Number
008 A;(14 / “7”)-((“01/01/”:(14(DY)))(DI) / ”7”)
009 R
010 6
```

Based on a tip originally by Tom Phillips. IS

Do you have a Tech Tip to share? E-mail it to editor@intl-spectrum.com

CLIF NOTES: WHO CARES IF MULTIVALUE SURVIVES?

Continued from page 31

radical discovery — not all data is flat. Yes, I know, I know. The MultiValue community has been saying that for decades. And there seem to be a lot of people rather upset over that fact. Here is something that we have been preaching for almost 40 years, and along come these upstarts with their degrees and fancy ideas and terminology, restate one of our main points of view, and are starting to take portions of the application space by storm. And they are getting all the credit!

You know something? You can accomplish a lot in this world if you don't care who gets the credit. For those of us who are in the business of running businesses, we would much rather have continuing and larger revenue streams. And because our MultiValue technology is something that we have proven allows us to develop solutions to problems quickly, we should be able to do it at a lower cost. Higher revenues, lower costs… that smells like profit to me.

Of course, that is not going to happen in the current millennium with green-screen, isolated from the web, won't talk to anything else applications. I think MultiValue is well-positioned to allow us to start producing breakthrough applications that integrate well in a heterogeneous environment, talk all of the modern "mainstream" protocols, and deliver data in whatever format and on whatever platform customer demand dictates. But in order to do so, our current crop of MultiValue professionals need to upgrade their skills.

First, we need those skills to start producing these kinds of applications. Otherwise, the MultiValue systems already in place will continue to be viewed as obsolete dinosaurs to be kicked to the curb within two years.

Second, we need seasoned professionals who are fluent in the modern technologies to help train newcomers to MultiValue. You can't expect to attract, educate, and retain new, young, top-notch talent if nobody in the MultiValue group in your shop even knows that the keyboard has a Caps Lock key on it.

And third, who wants to keep doing the same old thing year after year, decade after decade? How boring. Change is one of the things that makes Life interesting.

So this year, how about learning a new programming language and how to use your platform's APIs from that language to access MultiValue data? Learn to use your platform's tools to produce and use web-services. Explore mobile platforms. Come to the International Spectrum Conference this April to find out all of the neat things that other MultiValue professionals are doing with the database and tools.

As for me, I can't see just doing what I've done for the last 37 years for the next 37 years. (No, I am not one of those people who sits around dreaming of retirement.) So one of my goals for 2012 is to continue to dive into these new technologies deeper than I have already. I want to assist companies with MultiValue applications drag them into the 21st century and continue to expand them to support whatever business environment evolves from new, exciting technologies and customer expectations.

So I recommend that the MultiValue community stop worrying about whether or not MultiValue will survive. Jump in, look to the future, and press ahead. If we concentrate on "Thrive," the "Survive" will take care of itself.

That is, unless the doomsday predictions for next December when the Mayan calendar expires turn out to be true. But then, it will be kind of a moot point. I don't put any stock in those predictions. But sometimes in the wee hours of the dark, your brain will randomly whisper, "What if?" What if it is true? What if I am one of the 500,000,000 to unfortunately survive?

Me and 499,999,999 singing munchkins. Now that is a nightmare. **IS**

Clif Notes

Who Cares if MultiValue Survives?

BY CLIFTON OLIVER

Before we explore this question, let me get one thing out of the way — the obligatory New Year's resolution.

I will not write controversial column titles.

There. Done. And all in keeping with my tradition of starting the new year off with a burst of productivity by simultaneously making and breaking my New Year's resolution.

With that out of the way, let's return to the question. Who cares if MultiValue survives? I don't.

Oh my, I can already hear the scraping of sharpening stones on steel, the squawking of fowl, and smell the odor of tar being heated in the pots. I should hasten to explain myself before it becomes necessary for me to go on Amazon.com and place an order for large quantities of engine degreaser and burn ointment.

I started working with MultiValue around 1974. At that time, I was informed that it was probably not a good career choice to get too deeply involved with this product since it was "obviously" going to be completely obsolete and totally disappear by 1980. Of course, that didn't happen. It's lucky for me that I ignored such advice. MultiValue has not only provided me with a living for about 37 years, but just think of all of those International Spectrum Conferences, exhibits, technical sessions, dinners, parties, and hospitality suites I would have missed out on if I had thrown in with the COBOL programming and CODASYL crowd.

Then of course, there was the relational database and SQL movement. To say that this was phenomenally successful would be an understatement. As you would expect, the predictions of the imminent demise of MultiValue was rampant. If you were not fully relational and supported ANSI SQL as the primary data definition, manipulation, and query language, you were headed for the trash heap. Forget the fact that the Real World does not follow a precise mathematical model. When the only tool you have is a hammer, every problem begins to look suspiciously like a nail. If your application's data did not readily conform to the idea of tables and full normalization, change your application. If the resulting application did not support the unique way you do business, why just change the business. This bass-ackwards way of thinking, of course, led to the success of databases like Oracle and DB2.

It was also responsible for millions of dollars being spent on application redesign and platform conversions that in many cases failed and in some cases destroyed the companies attempting them.

"So how long is two years in your company?"

(For you relative newcomers, this is one of the ongoing jokes in the MultiValue community. Almost every time we hear of one of these attempts, it all started with an edict coming down that, "We are going to be off MultiValue in two years.")

And yet, MultiValue survives.

Of course, some MultiValue companies have gone out of business, some have merged, some have been acquired. You know, the normal evolutionary trends of businesses in a pseudo-free-market environment. And every time we see a change like this, a group of MultiValue munchkins starts marching around the message boards, newsgroups, and other areas on the Internet singing and chanting, "Ding, dong. The database is dead!" (Apologies to Wizard of Oz author L. Frank Baum.)

I'm still waiting.

And now, lo and behold, we have the NoSQL movement moving into prominence. They seem to have made a

Continues on page 30