


INTERNATIONAL Spectrum

THE MULTIVALUE TECHNOLOGY MAGAZINE | SEPTEMBER/OCTOBER 2014



Understanding Semantic Web

We Talk to the Experts



ALSO IN THIS ISSUE

- ◆ UniVerse and UniData Hashed Files – Part 2
- ◆ Memory, the Other Kind
- ◆ The Why of Python. The Why and Why Not of JavaScript

April

13 - 16

2015

The Premier MultiValue User Event



From application development and technical systems support to complex business and web development, we offer complete IT solutions.

RETURNING TO THE PGA RESORT, PALM BEACH GARDENS, FLORIDA



The **International Spectrum MultiValue Conference** is the only place that provides diverse solutions and support for integrating and developing your Enterprise applications into the world of business and technology.

This annual event is the place to find the **knowledge**, **people**, **resources**, and **technologies** for enhancing your MultiValue (D3, UniVerse, UniData, Reality, jBase, Caché, QM, and OpenInsight) software applications.

Topics covered at the International Spectrum MultiValue Conference include:

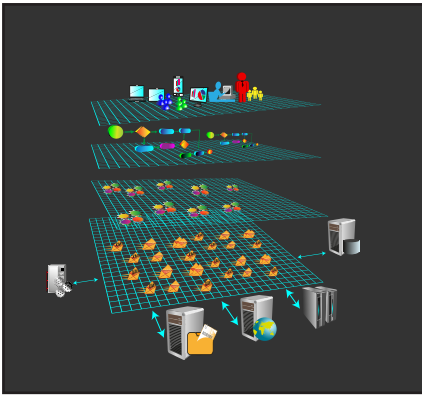
- **Integration**
- **Database Management**
- **Mobile Integration**
- **Reporting and Data Presentations**
- **Project Management**

...and more.



REGISTER AT

www.intl-spectrum.com/conference



6 Understanding Semantic Web -

Experts Q&A David Amerland talks to us about moving the focus from data to definitions. He'll give us a better understanding of how the Semantic Web (Web 3.0) relates to our existing business data and systems.

While he brings an Outside-of-MultiValue view, we also get a perspective from our MultiValue experts about designs and technologies that can be used with the Semantic Web.

FEATURES | SEPTEMBER/OCTOBER 2014

9 **UniVerse and UniData Hashed Files - Part 2** In part two of this series, Peggy and Jeff will guide us to a better understanding of the hashing mechanism, from modulo and separation, all the way to overflow and resizing. **BY PEGGY LONG AND JEFF FITZGERALD**

17 **Business Tech - Computer Memory...the Other Kind** Sometimes we need to take a fresh look at what we do. While our ground-level perspective is superior for detail, we need to pull back a few thousand feet to really see how things connect. This installment of Business Tech asks the deceptively simple question "Is your data complete?" **BY CHARLES BAROUCH**

20 **What Should I Learn Next? - JavaScript and Python** Rocket has picked Python as an important alternative to BASIC. Find out what Python brings to the table, and see why you might be adding 'Python Programmer' to your resume. **BY BRIAN LEACH**

DEPARTMENTS

From the Inside page 4

From the Press Room page 13

International Spectrum and MultiValue are registered trademarks of International Spectrum, Inc. All other registered trademarks are the property of the respective trademark holders.

From the Inside

"There is nothing wrong with your existing MultiValue Line of Business (LOB) software! There is everything wrong with your LOB 'user experience'."

This is not just an issue with 20-30 year old MultiValue applications, but all the 20-30 year old main-frame/client server LOB software. The LOB software is stable, or your business would have bankrupted by now. It is customized to make your business run smoothly. It is efficient when generating responses to specific business reporting and Business Intelligence (BI) questions. It validates, stores, and responds to the users...

And the users hate it!

If you listen closely to which users are making the most noise, it's not just your C-Level management, but your end users. It is those Millennials; all your newly hired employees.

Why? Their expectations are different because they grew up with technology. They have seen technology evolve quickly from desktop computers, to laptops, to smart phones, to tablets, to phablets. All these tech solutions have caused the users' interfaces and software experiences to evolve beyond what software developers could have envisioned as few as 10 years ago.

So what?! Those are all things used by individuals; consumers. They are comparing Microsoft/Google/Apple/Intuit's software offerings to your hardened business application. We know they aren't the same.

And we are right. Consumer grade applications are not the same. They set the expectation. In the past, LOB software developers could "blow off", or "work around", these

user expectations by providing dashboards and prettier reporting. When C-Level management had the only complaints which mattered, we could focus on just the points where the bosses interacted with the systems and data.

Nowadays, the end users are just as (and sometimes more so) tech savvy than the IT department. This causes "rogue users" to show up more and more often. These are your company's employees, deciding they can do it better, or thinking IT is too slow to do what they want and need. I won't go any deeper into the issue of "rogue users" right now, other than to say that they are the reason that everyone is screaming for change.

This change has nothing to do with the LOB software, and everything to do with the interfaces and experiences users expect from the LOB software. They see QuickBooks, and ask: "Why can't the LOB software do that?" They see Salesforce, and ask: "Why can't the LOB software do that?" They see Google Apps (Sheets, Maps, BI) and ask: "Why can't the LOB connect to that?" They look at Outlook and Office and wonder: "Why can't the LOB integrate... everything else seems to connect to Office?" They see iPhones and Android phones and wonder: "Why can't the LOB be accessed from them?"

It's not about GUI. It's not about Dashboards. It's not about desktops and tablets. It's not about Mobile, and it's not about web.

It's about interaction. It's about data. It's about information, and about using the information their way, when they need it.

When you have users who are just

NATHAN RECTOR
President

CHARLES BAROUCH
Editor

SYDNEY BAROUCH
Editor

TRACEY RECTOR
Layout



Learn more about the MultiValue Symbol and see what MultiValue 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. Copyright 2014 International Spectrum, Inc. All rights reserved. Reproduction in whole or in part, without written permission, is prohibited.

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.

International Spectrum is a registered trademark and MultiValue is a trademark of International Spectrum, Inc. All other registered trademarks and trademarks are the property of the respective trademark holders.

as tech savvy as the IT department, the users no longer ask for help to figure out how to get something done. They just do it until something blows up in their face and you are required to fix it. Again, this is a separate topic, for another day. This article isn't about the "Modern IT" department, and how the IT department is changing.

The key thing to keep in mind here is that your users want to interact with your LOB data and software, and not be controlled by it. Most LOB software is very controlling. You can only access the LOB data one way. You can only get information into the LOB software one way. You can only interact with your day-to-day contacts, accounts, and process with one interface.

Your users want to interface with the LOB from any device, software, or interface they have, especially the ones they use the most. This means, if they are in front of Outlook all day, they want to interface with the LOB from Outlook. If they want to look at reports, then they might

want it in Excel. And they want any changes made in Excel to be processed through the LOB software for bulk updates.

If they aren't currently at their desk, they want to access the LOB from their smart phone and tablet. If they are at their desk, then they want to access their information via a larger screen and keyboard. These adaptive interfaces are not limited to what hardware is being used, but to the environment the user is in.

Are they checking in from the airport? Are they talking on the phone while commuting to work? Is this a quick look during lunch or dinner? Are they on the warehouse floor or manufacturing floor? At their desk? On the web? Are they answering email? Are they sitting at home?

You can no longer create just one interface to your LOB application. Each of these interfaces change how the user wants to interact with the data, which in turn affects the subset of the data they wish to see. How someone uses LOB data and soft-

ware on a smart phone is different from how they use it on tablet, even though both of these are mobile devices.

If they have a keyboard and mouse, a user does not want to reach up to touch a screen, they want to keep their hands on the keyboard and mouse. If they are at a customer's location, then they want to be alerted to information that is useful to them in context – like low inventory or recommended sales – not all the other information, like internal company communications.

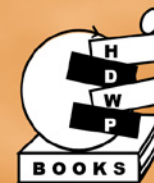
All these expectations are focused on how the user is experiencing the LOB application, not on how the LOB does *its* job. It's about how the user can do *their* job to help the business run.



NATHAN RECTOR
President
International Spectrum
nathan@intl-spectrum.com



When was the last time you read something fun?



Bottom line: *Tiago and the Masterless* is a book that only a programmer could have written, but not one that only a programmer can love.

- Jon Frater's review on Amazon.com

Upcoming
Tiago in the Tunnels of Krall
Tiago faces the Abyss

<http://www.hdwpbooks.com/books/tiago>



Understanding

Semantic Web

Experts Q&A

Just as XML gave us a common framework for exchanging data, Semantic Web seeks to create a common way of seeing information so that disparate data can be exchanged without custom interpretation being required for every pair of systems. In other words, Semantic Web is about semantics — the consistent usage of words — more than it is about data, per se.

XML is, as those of us who work with it know, still evolving. Semantic Web, likewise, is not a finished, packaged entity. We are still in the early stages of this technology. Despite that, it is already being deployed to great advantage.

The original definition comes from Tim Berners-Lee: *I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web — the content, links, and transactions between people and computers. A “Semantic Web”, which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy, and our daily lives will be handled by machines talking to machines. The “intelligent agents” people have touted for ages will finally materialize.* [Source: Wikipedia]

The problem I see with anything resembling AI is that we human types tend to think in terms of language when intelligence is iconic.

For a better understanding of this emerging technology, we spoke with David Amerland. David is a Fortune 50 consultant on Semantic Search and has authored several books, including [Google Semantic Search <intl-spec-trum.com/s/19z18>](http://Google.com/semantic-search-intel-spec-trum.com/s/19z18).

Outside Experts Interview

IS: *Semantic Web is just about search? Or is there more to it?*

DAVID: There's another layer to the Semantic Web. It's happening primarily behind corporate doors. Because they are beginning to see the value of understanding better what's happening within [the data]. When the great credit crunch happened, we realized a couple of things. Despite all the safeguards, no one knew who owned whom or what or where. Since then, the finance industry has launched an initiative, which is quite broad,

in which it applies new abstraction data formatting to the information of company entities; which allows us to identify other relational connections amongst them. So, three years from now, if we have another credit crunch, we would actually know exactly what company X owns in terms of other companies, and who owns whom, and how the connections are actually being made. Which is a big positive thing. And this is quite opaque. It's happening behind finance corporate silos.

IS: *Let me stop you there for a moment, because while I understand the idea of integrating data to find those kinds of connections, I'm not sure what makes Semantic Web a specific technique for doing that. Could you talk about what makes that a particular approach or methodology? What defines it as Semantic Web?*

DAVID: Well, we always have the ability to interrogate a database. What we didn't have was the ability to interrogate databases across the entire industry sector, or even, the entire world. And this is exactly the direction that Semantic Web is walking towards. And I say walking, because it is not very fast in, and it is an incredibly painful process. But, it's getting to the

stage where we're getting to that direction, where the moment it happens we see a couple of things. First of all, the traditional opaqueness which happens behind – which is created by compartmentalization of silos – which keeps information locked up in certain silos, will go. Information travels a lot more freely. And it actually makes sense as it gets out of where it is and goes somewhere else. And the other thing which we will see is that the moment we get that, we actually begin to see a meta-layer of connections. If we see, for instance, a piece of information which is held in place A, which then has an impact on another piece of information which appears to be separate on point B, suddenly that connection begins to mean something which we can infer from that. So, by joining these dots, by joining these data points, we begin to see how the world works at a much more grainy kind of way; which allows us to then operate in a different manner.

IS: *So then the thing that makes Semantic Web interesting and important is its ability to take unstructured data and make clean assumptions about it? So that it can link two disparate pieces of data? Is that a correct summary?*

The full interview can be found here <intl-spectrum.com/s/19z1e>.

Inside Experts

While we interviewed David as an expert from outside the MultiValue Industry, we also sent out a survey to our MultiValue software providers and VARs, soliciting their input.

QUESTION: *Semantic Web starts by developing rules for common definitions. What 'odd duck' cases have you encountered? For example: one system counts dollars as*

all monies received, but another system counts taxes separately, so that the word "dollar" actually has a different meaning on each system.

HDWP: We had a system where they thought they were storing all times as US Eastern. It turned out that they were storing time based on the location of the data entry. So, if New York put in a Dallas ticket, it was US Eastern, but if Dallas did their own data entry, it was showing as one hour earlier. The more offices they added, the worse the mess got. We finally had to split the entries by office and cross them by data entry person. Where office and entry person agreed, we auto-fixed it to UTC. Where they contradicted, we checked the records and fixed them by hand.

PRECISION SOLUTIONS:

You bring up a good point about the common definitions. Heck, even getting people to agree on the standard abbreviations for units of measure, currency, or country codes can be a challenge – and these are established ISO standards! We deal with this in EDI all the time; our customers call a case of materials a "CS" and the trading partner calls it a "CT" or even worse "A0". So it's more fundamental even than "dollar", which is of course one of many currencies with different decimal precision and valuation rules. (See Euro vs. GBP for a case study in madness.)

QUESTION: *Semantic Web builds on these 'a dollar means a dollar' algorithms by adding an intelligent search layer. What sort of tricks have you used to enhance search on systems you work with or develop?*

HDWP: We built a pre-processor to create a soundex-variant of the descriptive data. Then, when someone typed

a query, we ran their string through the same soundex-variant before we did the compare. Someone looking for the words "bread crumbs," would have their search string converted to "BRBSRMS" <see chart below> which would match the pre-processed search strings.

```
b → B
r → R
e → <null>
a → <null>
d → B
c → S
r → R
u → <null>
mb → M
s → S
```

PRECISION SOLUTIONS:

Assuming there are established and accepted standards for literally everything, searching for a "dollar" is a remarkably American search, just as regional as searching for "flore". Searching for a Euro is broader, and I suppose at some level if we classified dollar, euro, and floren as currencies, a system could automatically look up the currency exchange to present a "value" in a local currency or a variety of currencies. This would cause the user to see the current – virtual – value of their currency, not the stored value.

QUESTION: *Semantic Web is envisioned as an underlayer for Expert Systems or Decision Support Systems. What sort of work have you done with Expert Systems or Decision Support Systems.*

HDWP: We had a client who had an array of different sized boxes available for shipping. In their rush to get orders out, the clerks would often grab the nearest boxes instead of looking of the most efficient fit. We wrote an algorithm to determine the optimum fit for each order. We printed the packing slip and the box labels based on it. Taking the guesswork out of the packing

Quality Counts

PASS

FAIL

mvTest

<http://www.brianleach.co.uk>

made the job faster, so the clerks used the pre-generated rules. The end result was fewer boxes going out half empty and orders going out with the smallest number of separate boxes. This made packing up the truck more efficient. It also made delivery faster because the driver had to locate fewer boxes at each delivery stop.

PRECISION SOLUTIONS:

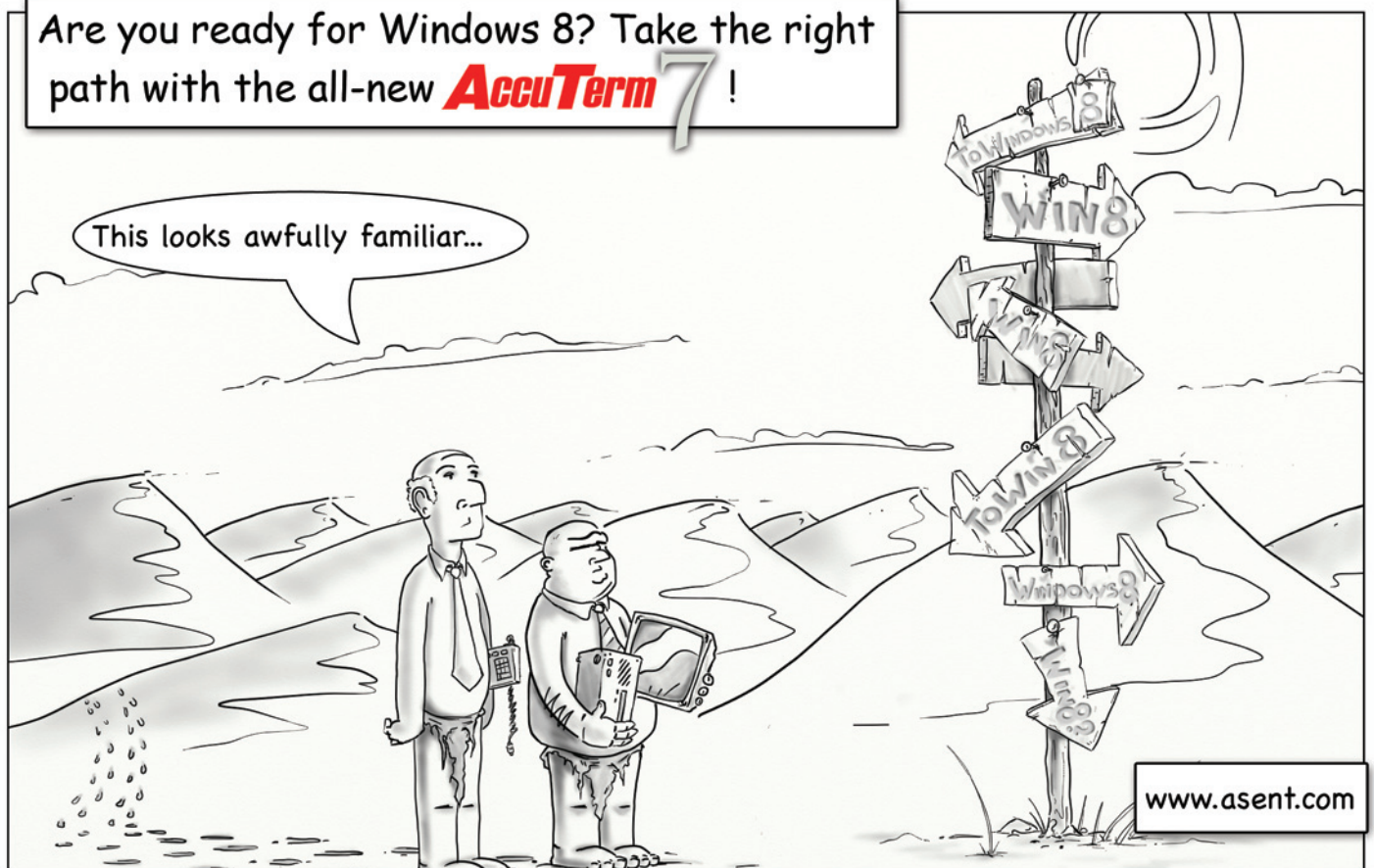
The problem with AI, however, and the problem I see with anything resembling AI, is that we human types tend to think in terms of *language* when intelligence is *iconic*. Take for instance the word “hand”. Is it a noun? Is it a verb? One needs to know what it is to know how it relates to its surroundings. Even as a noun there are multiple independent concepts that the word might describe. Each of these concepts is iconic unto itself, and how that icon relates to its surrounding icons is itself

iconic. However, when we think in terms of language, words like “has”, “is”, “near”, and “far” are too ambiguous to define atomically and therefore establishing any level of definiteness is problematic. Without a concrete foundation, the model collapses under the ambiguity.

Simply providing machine readable information and standards is one heck of a good start. But it does little to establish context and iconic relations, which are essential for comprehension. Without comprehension, there is no understanding, and without understanding there is no intelligence, machine or otherwise. Maybe I’m just a pessimist, but from all I’ve read, it seems that people have approached this problem all wrong for a good long while now. **IS**

Are you ready for Windows 8? Take the right path with the all-new **AccuTerm 7**!

This looks awfully familiar...



Hashed Files

Part 2

BY PEGGY LONG AND JEFF FITZGERALD

Hashed Files are one of the key features of a Multi-Value Database that make it efficient for business applications. In our last article, we talked about how the databases generate the pointer or location to a record using a hash key.

In this article we will explore how the record data is stored in the hashed file's groups.

Data Storage Within the Group

At this point, we've explained how data records are assigned to groups; now let's have a look at the way in which the data are stored in the group. Because of the variable length nature of the data, the records are treated as a list. The most significant implication of this is that locating a specific record within a group requires a scan of the records within the group. This has major performance implications that will occupy much of our series of articles.

UniVerse and UniData organize the data records within a group in very different ways. We will describe each method, beginning with UniVerse. Our description will start with the basics, ignoring exceptions such as large records. Later articles will refine the description to include the subtleties.

UniVerse treats the records in a group as a linked list – each record has a link

UniVerse and UniData organize the data records within a group in very different ways. Our description will start with the basics, ignoring exceptions such as large records. Later articles will refine the description to include the subtleties.

to both the next record and the previous record. There is a twelve-byte header at the beginning of each record. Those twelve bytes contain the forward and backward links and a number of bits that reflect various pieces of status information about the record. Immediately following the record header are the record key and the record data, separated by a segment mark (the HEX "FF" character). The forward link is the physical address – the byte count within the group – to the beginning of the next record. This facilitates scanning the group because it provides the address for the next read. It also facilitates data retrieval because subtracting the current record address from the next record address yields the length of the current record.

UniData uses a different method, in which the record keys serve as an index to the data locations. A position near the middle of the group is chosen – we will refer to this location as DATAPOS. The data record keys for the group are placed in a list, with the number of characters in each key preceding the key. For example, if our record keys were ONE, TWO, THREE and FOUR the list would look like this: "3ONE3TWO5THREE4FOUR". BUT, the list is reversed – it starts at DATAPOS and goes backward toward the beginning of the group, so it would appear more like this: "FOUR4THREE5TWO3ONE3". Starting at the beginning of the group are pairs of numbers containing the length of each record and the physical address of the data for that record. The data itself is written after DATAPOS and grows toward the end of the group with a segment mark between the data for each record.

This approach offers some advantages, too. Scanning the list of record keys for a particular record is quick because we are simply parsing a short string of keys – there is no need to read past the data. Once the required record is located we get the corresponding length / address pair which tells us how much data and where it begins.

Buffer Size, Blocksize and Separation

After reading the preceding discussion of how data records are stored in groups, a natural question is: “How big is a group?” The answer is: “That depends.” The size is configurable by the user at the time the file is created. Both UniVerse and UniData allow specification of the buffer size used by the file, however, the two environments use different terminology.

UniVerse uses the term “separation” to define buffer size. Separation is the number of 512 byte units that compose a buffer. So a separation of 1 means the buffer size is 512 bytes; a separation of 2 yields a buffer size of 1,024 bytes; separation 4 gives a buffer size of 2,048 bytes and so forth. The syntax of creating a file varies in UniVerse according

to the “account flavor” being used. In the “Ideal” flavor, the command looks like this:

```
CREATE.FILE filename type modulo separation
```

For example:

```
CREATE.FILE TESTFILE 2 3 4
```

This will create a file named TESTFILE with a file type of 2, a modulo of 3 and a separation of 4.

UniData uses the term “blocksize” to indicate the buffer size of the file. The UniData commands use a “blocksize multiplier” to specify blocksize. A blocksize multiplier of 0 produces a buffer size of 512. Positive integers from 1 to 16 are multiplied by 1,024 so that 1 gives 1,024, 2 yields 2,048, etc. The highest buffer size available in

UniData is 16K. The UniData command to create a file looks like this:

```
CREATE.FILE filename modulo, block.size.multiplier TYPE type
```

For example:

```
CREATE.FILE TESTFILE 3,2 TYPE 0
```

This will create a file named TESTFILE with a modulo of 3, a buffer size of 2,048 and which uses hash type 0.

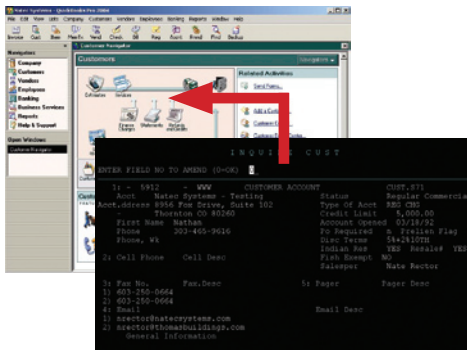
Future articles will deal in depth with strategies for picking buffer size and will explain the interactions and performance implications of various choices.

Overflow

Now that we’ve discussed the size of buffers used in groups, it leads naturally to the subject of “overflow”. Since the buffer size is finite, what happens

mv
QB

QuickBooks API for the MultiValue Database



- **Read/Write Directly to Quickbooks Databases**
Customer, Vendor, Invoices, Purchase Orders, Chart of Accounts
- **mvQB API is Designed for the MultiValue Program to Use**
All routines are simple BASIC calls designed for the developer. No special user interfaces required.
- **No Need to Learn the Internals of QuickBooks**
- **QuickBooks Pro/Premier/Enterprise**

NATEC
Systems



Providing Solutions to your MultiValue Questions

Phone: 303.465.9616
E-mail: mvqb@natecsystems.com
Website: www.natecsystems.com

when more data is hashed to a particular group than can be contained in one buffer? For instance, suppose that we have a UniVerse file with a separation of 4, thus each group starts out with one buffer of 2,048 bytes. Suppose we add 30 records that average 100 bytes in length and all 30 records happen to hash to the same group of the file. The amount of data to be stored is 3,000 bytes – more than can be held in a buffer of 2,048 bytes. What now?

Both UniVerse and UniData accommodate this situation by adding “overflow” buffers to the group. An overflow buffer is an additional buffer that is linked to the primary buffer of the group. From a logical point of view the group can be treated as though it were simply extended and doubled in size. From a physical point of view, the overflow buffer is not contiguous with the primary buffer and additional disk

I/O will be required to retrieve it. The more disk I/O that is needed to retrieve data the slower the retrieval will be. Since there is more data in the group, the process of scanning for a required record will also impact performance.

If the first overflow buffer becomes full, additional overflow buffers are linked to the group. In fact, a group can be composed of as many overflow buffers as are required to hold the data. But the longer the chain of overflow buffers in a group, the slower access to data in the group will be. And the relationship between overflow and performance is not linear – it is geometric. This means that retrieving data in the tenth overflow buffer will be more than twice as slow as retrieving data in the fifth overflow buffer.

It is safe to say that overflow in hashed files can be the single biggest factor in system performance.

Resizing

Since overflow has such a huge performance impact, you might guess that there are strategies for eliminating or reducing it. Both UniVerse and UniData provide tools to resize their files. Resizing recreates the file with altered values for the modulo, file type (hashing algorithm) and buffer size. Intelligent choice of these parameters can turn a poorly performing file into a fast one.

Resizing is time-consuming and requires downtime. The analysis and selection of optimum parameters is not always straightforward. Future articles will talk in greater detail about the trade-offs and strategies for file tuning. The hashed file basics that we’ve explained will make these future discussions more readily understandable. **IS**

10 Years Strong & Growing

Integrity, Efficiency, Service,
Guarantee, Global Productivity,
Midwest Rates, Universe,
UniData, D3, Barcode Services,
Nationwide Support, Web Services,
Custom Barcode Programming,
.Net Programming, Mentoring,
Project Outsourcing



www.pickprogram.com
contact@pickprogram.com
(614) 921-9840

Do you... have projects sitting on “the list” and not being completed?

Do you... just need some assistance from time to time?

Do you... want up-front, guaranteed estimates?

Senior Consultants are Currently Available.

“PICK Programmer’s Shop has been outstanding to work with. They work with the highest integrity, are very responsive and quite knowledgeable. I can’t imagine operating without them.”

Patti Rowlette, Rowlette Executive Search

It’s not just an anniversary. It’s a guarantee.

Thirty years ago PEGGY LONG and JEFF FITZGERALD created FAST, a utility which can quickly scan a file, report errors and recommend the optimum MODULO and TYPE parameters needed to RESIZE a file.

Their thirty plus years as file maintenance and performance specialists makes Peggy and Jeff uniquely qualified to speak on this subject. Additionally, Jeff brings his experience as a police detective (10 years) and Peggy brings her knowledge as a Ph.D. in Music Education.

Their company, Fitzgerald & Long, Inc. is located in Colorado. If you have questions or comments, please email them at support@fitzlong.com



Spectrum University Webinar Series

Upcoming Schedule

**International Spectrum 2015
Conference Preview**
Tuesday, October, 21 2014

**Windows 8 Metro Development for
MultiValue**
Tuesday, October, 28 2014

Calendar Data on the iPhone
Tuesday, November, 04 2014

**Developing MultiValue GUI
Applications**
Tuesday, January, 13 2015

**Responsive Desktop Applications with
MultiValue Data**
Tuesday, January, 20 2015

Spectrum University offers training classes in a variety of topics pertaining to the MultiValue market. Looking for training for yourself or employees? Spectrum University Can help! Check out our latest offerings.

- Weekly Live Sessions
- Only 60 minutes of Your Day
- Expert Presenters from the MultiValue Industry
- Only \$175 per connection
- Watch Individually or with a Group
- New Topics Each Week

Have a topic you want covered?
E-mail Nathan with your suggestions at
nathan@intl-spectrum.com

REGISTER AT

www.intl-spectrum.com/webinar

Presented by



FROM THE PRESS ROOM



New Product Release Introduces The Next Phase Of Integrated Ecommerce

Kore Technologies announced its new KommerceServer eCommerce Suite Release 5.3; an integrated solution for B2B and B2C websites. The new release features a redesigned content management system, new security features, new shipping and credit card options and advanced document management.

KommerceServer is designed to integrate to any back office system with Kore's Kourier Integrator product. The integration reduces data errors, saves time and reduces labor costs for web administrators. "We strive to provide a blended solution, with seamless back office integration and flexible content management capabilities. This results in a rich user experience and more effective sales channel for our clients," said, Co-Founder and Managing Partner, Ken Dickinson.

KommerceServer eCommerce Suite includes a robust webStoreFront, a webPortal and web Administrator module, as well as a Mobile

edition. Its framework and content management system provide support for business-to-business (B2B) and business-to-consumer (B2C) environments.

Kore Technologies was founded in 1999 as a software development and services company. Today, Kore is a leading provider of enterprise integration, business intelligence and eCommerce solutions for MultiValue and non-MultiValue databases.

For more information about Kore Technologies visit www.koretech.com or call 866-763-KORE. ■



Total Computing Solutions Introduces Snap Webstore - Android Tablet-based Point of Sale product for MultiValue Systems.

Total Computing Solutions (TCS), a leading provider of payment processing, point-of-sale and data security solutions, is pleased to introduce Snap Webstore a Point-of-Sale system with Android Tablet-based mobile Point-of-Sale

devices. As with other TCS products, Snap Webstore is implemented using MultiValue database architectures for reliability, security and performance.

Snap Webstore allows retailers to combine inexpensive, portable Android point-of-sale devices with the impressive capabilities of the TCS retail software suite. Retailers can now have the advantages of portability and simplicity along with the comprehensive inventory management, secure transaction processing, and extensive feature set they have long come to expect from TCS.

As with all TCS point-of-sale products, our TotalLink 3.0 P2PE payment processing system is completely integrated with Snap Webstore. TotalLink 3.0 provides the highest level of security for cardholder data from the actual card swipe all the way through any networks and systems until it reaches the payment card processor. Unencrypted credit card information is never stored or transmitted on your system.

Retailers using MultiValue systems can now use a full range of highly secure point-of-sale devices, ranging from a single portable Android tablet up through highly complex multi-store retail environments supporting hundreds of registers. All of these devices are supported by the same powerful inventory control, retail management and consolidated reporting features. In

addition, the Snap Webstore supports a comprehensive e-commerce store that is seamlessly integrated with the retail store.

Snap Webstore provides the easiest path to a comprehensive leading-edge retail strategy incorporating completely integrated point-of-sale, e-commerce, accounting, inventory control and retail management components. TCS helps you eliminate the hassles of trying to plug together different systems to meet the needs of your customers.

"Snap Webstore changes the game for retailers. Now they can be just as leading-edge as their latest competitor without abandoning the solid business management principles that got them where they are. Our Android tablet implementation gives them simplicity and ease of use, with a secure back-end system that is 'combat-tested' for real business." stated Jim Hammons, Vice President of Total Computing Solutions.

About Total Computing Solutions

Total Computing Solutions, LLC. provides leading Point of Sale, Payment Processing, and data security services for systems integrators and retailers across the country. With a wide array of customers and partners nationwide, TCS provides a comprehensive suite of products and services that provide extremely reliable

FROM THE PRESS ROOM

and highly secure retail and customer service solutions for our clients.

For more information on TCS and on Snap Webstore, please contact Total Computing at (801) 796-7600, visit the TCS website at www.total-computing.com, or send an email to sales@total-computing.com ■



Entrinsik Informer Rated Highly by Users in New Report

Entrinsik Inc. (www.entrinsik.com), developer of award-winning data management software, has announced the publication of a third-party report by TrustRadius (www.trustradius.com) summarizing user reviews of its business intelligence and reporting solution, Entrinsik Informer. TrustRadius is the leading online community for professionals to exchange candid insights and opinions about business software through in-depth user reviews.

The TrustRadius product report was based on 64 in-depth, authenticated user reviews about Entrinsik Informer on the TrustRadius website. The report also includes information from interviews with Entrinsik management

and independent research conducted by TrustRadius.

The report includes a company and product overview, product strengths and weaknesses derived from user reviews with quoted excerpts, aggregate user ratings compared to business intelligence product averages, and an interview with key management on product positioning and plans.

"Entrinsik gets very high ratings across the board, significantly higher than the BI product average across every reported facet by users," said Alan Cooke, Research Director, TrustRadius. "In particular, Entrinsik Informer rates higher than industry averages in likelihood to recommend and renew, product usability, implementation satisfaction, and support."

Doug Leupen, CEO and President of Entrinsik commented, "Our mission with Informer has been and continues to be to provide the most valuable, practical, and user-friendly reporting and BI solution on the market, and the many reviews of our product on TrustRadius confirm that. We continue to appreciate the outstanding support and loyalty of our customers."

The full TrustRadius report can be downloaded at www.entrinsik.com/TrustRadius. The full listing of Informer user reviews can be found on the TrustRadius website at www.trustradius.com/products/entrinsik-informer/reviews.

About Entrinsik

Entrinsik develops, implements, and supports Information Management Solutions that enable organizations to maximize performance and improve bottom lines. Entrinsik Informer is an innovative, award-winning agile reporting and business intelligence solution used by tens of thousands around the world. Entrinsik Enrole Registration & Course Management software drives many of the nation's largest continuing education profit centers and seminar event planning industries. Over 1,500 organizations around the world use Entrinsik's software every day. For a demonstration or a free trial, call 888-703-0016 or email sales@entrinsik.com. Visit www.entrinsik.com. ■

success. This is the sixth time that Zumasys has made the list.

For 33 years, *Inc.* has welcomed the fastest-growing private companies in America into a very exclusive club. The average company on the list grew a mind-boggling 516 percent. Those are results most businesses could only dream of.

As an *Inc.* 5000 honoree, Zumasys now shares a pedigree with Intuit, Zappos, Under Armour, Microsoft, Jamba Juice, Timberland, Clif Bar, Pandora, Patagonia, Oracle, and other notable alumni.

In addition to making the *Inc.* 5000 list, Zumasys recently ranked #1 on the 2014 Best Places to Work in Orange County list, a project of the Orange County Business Journal and Best Companies Group. ■



For Sixth Time, Zumasys Named to Inc. 5000 List of Fastest Growing Companies

Inc. magazine ranked Zumasys #3935 on its 33rd annual *Inc.* 500|5000, an exclusive ranking of the nation's fastest-growing private companies. This prestigious list of the nation's most successful private companies has become the hallmark of entrepreneurial

- Each four-drawer filing cabinet contains 8-10,000 pieces of paper and they each occupy 7.5 square feet of office space.
- It takes around 100 valuable staff hours to fill one filing cabinet, by the time it is full, many



The Issues with Conventional Paper Storage

FROM THE PRESS ROOM

documents will not have been retrieved since they were first filed.

- If a document is misfiled it's as bad as a lost piece of paper, in either case, the information is not available.

Surveys have found that, on average, searching for documents accounts for more than 4 hours a month for the average employee. Documents that have been lost must be recreated and this accounts for more than 6 hours a month. The cost of commercial property and the need to store documents means that paper based document storage competes with people for space within an organization.

UnForm Solution

Scanning documents and archiving them can greatly reduce the amount of storage space required by paper. Freeing up expensive floor space can result in large cost savings. The UnForm Document Management solution offers a very powerful search tool that will help you find documents quickly. You can search for documents by document type, creation date, keywords, or many other kinds of meta-information. Using OCR technology documents archived in UnForm can be automatically full-text indexed, which means you can find precisely what you're

looking for. You'll never again lose, misplace, or misfile your business documents.

Think about the time and money you'll save with UnForm. ■

the UniVerse and UniData product ranges, has now reformed as the Rocket Software User Group. This is to allow us to welcome other Rocket software users, including our MultiValue friends running D3 or mvBase. You can sign up at <http://www.rsusers.com>. ■



U2 User Group has Reformed as Rocket Software User Group

The U2 User Group that formerly represented users of

eXtra Newsletter

Stay on top of Industry News

- ◆ Tech Tips
- ◆ Job Postings
- ◆ New Products
- ◆ Corporate Updates



www.intl-spectrum.com/newsletter



GET CONNECTED.

KNOWLEDGE AND EDUCATION FOR THE MULTIVALUE PROFESSIONAL.

ABOUT OUR PROFESSIONAL MEMBERSHIP

We are all busy in our day-to-day work and staying up-to-date with the current MultiValue technologies can be difficult.

Professional Memberships provide you access to knowledge, solutions, information, and code that you won't find in other locations.

Professional Membership Includes:

- Magazine in Electronic and Print Formats
- Newsletter
- On-Demand Videos
- Live Webinars
- Discounted Conference Rates
- Research papers
- Case Studies
- Source Code



<http://www.intl-spectrum.com/membership/>

Computer Memory... the Other Kind

BY CHARLES BAROUCH

When we say “computer memory” we expect that people are thinking about ROM, RAM, maybe EPROM, or even firmware. That’s how we are trained to think about the term; as an aspect of the hardware. I’d like to put a more human spin on the term for this column. I want to talk about what the computer does, and does not, remember.

Show Your Work

In the simple electric calculators, “memory” becomes a content term, not a hardware term. It is the way we hold on to more than one numeric *thought* at a time. Imagine if we designed our primary business systems to work like a high powered version of the no-frills electric calculator: people key in data all day, but our machine can only retain one train of thought — one equation — plus a few completed values. Once the thought completes, and we add or subtract the remembered values, all we have is a result.

Tomorrow, when the boss asks for the total sales of all divisions over the lifetime of the business, we have the answer. If she asks any other question, we have to start data entry all over again

The computer has a memory of the order record that is accurate. We just weren’t using it.

because this sort of simple computer remembers none of the components.

Database, DUH!

In order to answer questions on the fly, we shove everything into a database. This is a database-oriented magazine, the readers are database professionals.... Why’d I even bring it up?

Because I keep finding system after system, using MultiValue and non-MultiValue data storage schemes, where the computer has not been taught to remember enough. Is this a sign of bad intent? No. Sloppy, stupid staff? No. It is a sign of people not building a set of completeness rules into the database design phase. We tend to — especially given the tight deadlines, small staffs, and real-world pressures — design to the current need. Unless we design with an eye toward the future, we have a problem when new questions are asked.

My crystal ball is in the shop. I do not have a sure list of upcoming needs. How do I future-proof my data as much as possible? I make sure it speaks in complete sentences.

Becoming Complete

Mark, our first customer, calls up and orders our bestselling product, the Fribbulator. The result of this one phone call is:

- customer record,
- ship-to record,
- order header record,
- order detail record,
- invoice header record,
- invoice detail record,
- packing slip head record,
- packing slip detail record,
- shipping record,
- A/R record of the payment.

Ten records, just to get this one order done. We must be complete, right? If the business closed today, we’d be 100% right.

However, we might be unfortunate enough to stay in business for another

three years. That is when Mark is putting in a second order. As he does, he tells us that his business has moved (i.e. they have a new customer address) to a different city, and the ship-to address is now the same as his new business address.

Our system has no time reference for customer data. Therefore the customer address — which has to be correct regardless of whether we are researching the new record *or* the order from three years ago — will screw up location-based reports whichever way I set it. Why? Because we don't store the shipping data and the billing data on the order, we store just the ID/link to the customer and ship-to <see figure 1>. The more our customers change offices, stores, and distribution centers, the more errors our database accumulates.

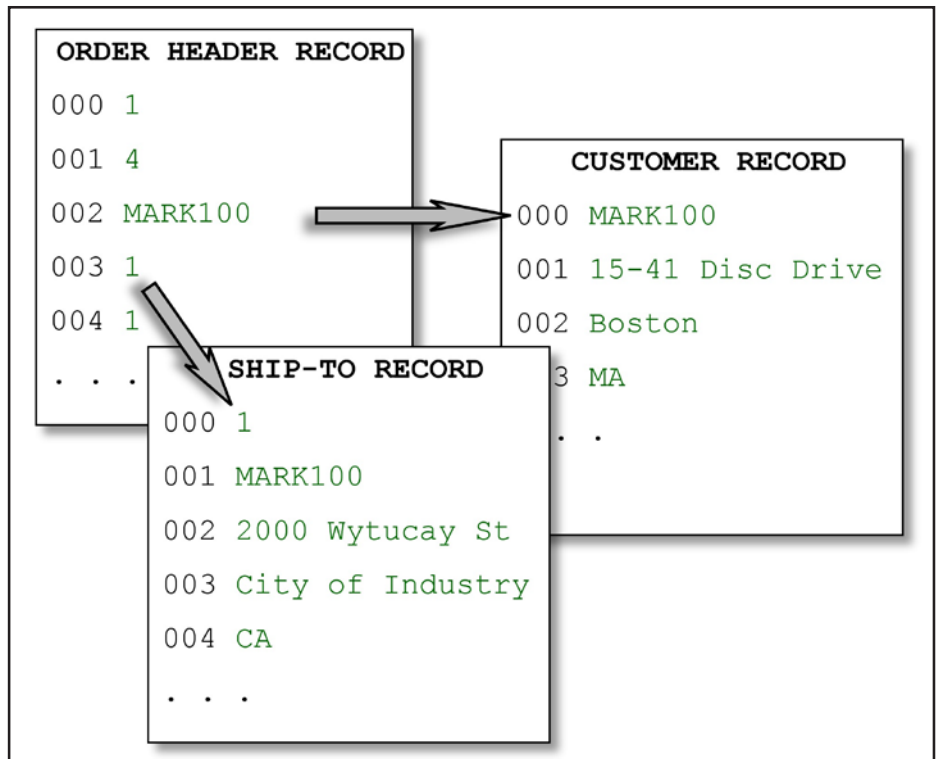


Fig. 1

We, hopefully, created a second ship-to record for the second order instead of allowing data entry to change the

existing ship-to. That saves us from half the problem; we know where each order went. For the other half — the



taking multivalued ...
where it has never been before

- Quick and easy to install
- High quality pdf documentation and online help
- Close compatibility with most other multivalued environments
- Maintenance-free file system for ease of use
- QMClient API for development of VB, C and web-based applications
- Very low licensing costs
- No mandatory support contracts
- AccuTerm bundled at no additional cost

Ladybridge Systems Ltd
17b Coldstream Lane, Hardingstone, Northampton, NN4 6DB, England
www.ladybridge.com
US Main Distributor: EasyCo, 220 Stanford Drive, Wallingford PA, 19086 USA
www.easyco.com

www.openqm.com

billing address — we have a shipping record. If we sent the billing address to UPS, we might be able to change our reports from the obvious choice (the customer record has the billing address) to the correct choice (the shipping record has it). That solves the rest of our problem.

We were saved because the computer has a memory of the order record that is accurate. We just weren't using it. Thinking about the completeness of the data makes us think differently about what the computer needs to remember.

Frighteningly enough, this isn't even a future-proofing issue. We didn't get hit with an unexpected question; this was one of the known ones: *How many Fribbulators did we sell in a particular city?*

You Complete Me

The completeness question which would have forced us to solve this example before it became a problem is: Can we protect our history data from future edits? Wow. That question is bigger than our fix. The complete answer would involve change control, too. It might involve software constraints to prevent certain data changes, such as editing the address on a ship-to.

Should I be allowed to edit the address? Yes, if there are no orders, invoices, or other transactions attached to it. No, if it is in use. Thinking about completeness, on something as (supposedly) dead-simple as an address, created a ripple through our entire design. Imagine the tsunami a full rethink might cause.

Not My Job

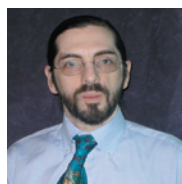
I know what you're thinking: Interesting, but I don't design databases or entirely new systems. I just patch the ones we have. This isn't about *my* job, right?

Wait... what if I build a completeness list and use it to test my current system? Instead of using it for design, I can use it to predict where the boat is secretly leaking, or might leak soon. This article is for anyone who designs, codes, or maintains databases.

What sort of completeness questions should we ask to do real future-proofing? Here are a few skeleton questions to get you started. Always think about your specific industry, and the unique rules of your particular company, and build up from there.

- When is it okay to see/change/purge/resell/re-use _____ data?
Example: Mailing lists
- Who has the authority to see/change/purge/resell/re-use _____ data?
Example: Payment method data
- What data can be left unencrypted?
Example: Marketing survey data (The answer might surprise you when you think about it.)
- Where do the needs of the business require us to relax data completeness rules? How do we keep partial data discreet from the complete data?
Example: Intra-company sales vs. customer sales
- Why do we collect _____ data?
Example: Archives of product descriptions

Willing to share your design questions? Want to tell us about processes you've had to re-imagine? We'd love to get a peek at how you think. Send your thoughts to editor@intl-spectrum.com. Thanks for the memories. **IS**



CHARLES BAROUCH
is the CTO of HDWP, Inc. He can be contacted at www.hdwp.com

Feedback

What came first, the letters or the letters-to-the-editor department?

International Spectrum Magazine has a Feedback Department, sometimes known as Letters to the Editor.

We want to hear your comments, your reactions, your agreement or disagreement with what you see. Also, do not hesitate to let us know about things happening in the MultiValue Community we may not have heard about yet.

Please send your comments by e-mail to:
editor@intl-spectrum.com

PICK PROFESSIONAL

Don't make the mistake of placing your career in the hands of just ANY search firm. We are professionals who understand the importance of CONFIDENTIALITY and RESPECT of a job search, and our database of clients is one of the largest in the country. Unlike the rest, we will work in YOUR best interests' to help you further your career. Because of our dedication and professionalism, we are recognized as the leaders in the PICK/UniVerse/Unidata placement industry in the Tri-State area and throughout the U.S. So if you are tired of putting yourself at the mercy of the rest.

CALL THE BEST! Contact...

Matt Hart

EXECU-SYS, LTD

1411 Broadway, Suite 1220
New York, NY 10018

(212) 967-0505

(800) 423-1964 x 302

Fax: (212)947-8593

Email: mh@eslly.com

Consultants Needed Nationwide

WHAT SHOULD I LEARN NEXT?

JavaScript and Python

BY BRIAN LEACH

So what languages are the hip young developers writing in today?

Why JavaScript?

JavaScript obviously is the first language that comes to mind. From humble beginnings, as a means of adding dynamism to web pages, it has escaped the tyranny of the browser. A good example of this is Node.js for server application development and Phonegap for native mobile development. They are both based around JavaScript as their programming language.

Additionally, JavaScript has an ever-growing wealth of frameworks and libraries which have ably expanded its power. It is proving to be not only the world's most popular language, but also, according to Google's benchmarks, extremely fast.

Because it is an interpreted language, it is easier to optimize. JavaScript can outperform even C# for many tasks. Douglas Crockford's excellent books <<http://javascript.crockford.com/>> have done much to awaken people's understanding of, and respect for, the inherent capabilities of this language.

Rocket's research points to this being a popular choice; cross-platform and easy to integrate and license. If you were going to add support for another language, which one would you choose?

JavaScript expertise has become, for younger techies wanting to show off their skills, a part of their bragging rights. For mass appeal, JavaScript is certainly more than just the flavor of the month.

Why Not JavaScript?

If you look a little deeper, you'll see that this is not the whole story. While JavaScript has a huge impact on the client tier, some impact on app development, and occasional forays into the middle tier through Node.js, it's not really statistically significant for other types of development. 'Real' programmers will still look to C# as the standard today (long gone are the days when C++ was used for anything other than geeky low-level code hacking, and few tears should be shed over that) and

Java sadly refuses to die in the corporate world. If you want to see where the interests of the bright young things of today's programming generation lie, you just need to pick up one of the raft of web or app development magazines. Behind the RWD frameworks, lifestyle consumer goods, jQuery tips, designs of the month and how-to-hack-WordPress articles, you will see (admittedly rare) examples of other languages at work.

Not so incidentally, this makes Rocket's recent announcements that their U2 products will be adding support for one of those languages - Python - all the more interesting.

Why Python?

Of course, nothing is new in IT. Right now, dynamic is hot. Currently, dynamic languages are very much the *in thing*, with two hairy old contenders - Ruby and Python - leading the field. These languages are made for an impatient generation who want quick and easy results. They are blessed with the horsepower to deliver to those who are not easily wooed by the cleverness of a short and cryptic C function. And that is not a bad thing, since focusing on

delivery is something that should strike a chord with many readers of this magazine - it is, after all, a chief benefit of our own technology niche.

Why Python? Well, from their stated point of view, Rocket's research points to this being a popular choice; cross-platform and easy to integrate and license. But then, by the same arguments, why not JavaScript, Ruby, Java or that other old workhorse, Perl? If you were going to add support for another language, which one would you choose?

JavaScript would seem to be the clear winner in popularity alone. But here's the rub: it is a fully interpreted, prototypical, enormously flexible language whose code is dynamically extensible at *runtime*. Halfway through an application you can choose to add or remove methods from an object, arbitrarily discover new code, and inject new properties. These are the very things that make JavaScript a perfect choice for the front end. However, they all ring alarm bells for DBAs.

The potential for error is just too high. Prototypical inheritance and extensible classes are not natural for a generation of developers who grew up with a more standard OO model. JavaScript code,

which works in very different ways depending on what has gone before, may be fine in the limited context of a browser or app. But across a large business system, that is the sort of headache that over-stretched support teams can do without. Java is owned by Oracle, so it is hardly a likely choice for another database vendor. C# under Mono would have been a nicer option. And as for Ruby - well that's just weird.

Python may not be every developer's favorite, but it is simple, easy to read, has object orientation, and supports procedural and functional models as well as structured programming. It does allow dynamically adding and subtracting methods and properties to classes, but that is possibly less prevalent with Python than with JavaScript. Libraries like Require.js, in JavaScript, allow the downloading of code on demand. This has evolved as a response to browser speed rather than a need within the language itself.

Python is *mature*, having first been designed in the 1980s, and *open* since 1991. It has hooks which allow you to extend it using C, as well as hooks to bind it into other environments. Iron Python, the version for .NET using DLR, increased its popularity with another, more skeptical, group of pro-

grammers and did much to inject new excitement around it as a scripting language. Because it is a dynamic, untyped language, it should work well with the vicissitudes of MultiValue data. The large number of extension libraries (or 'modules') bring a wide range of new functionality to the table.

But do we need it? Surely the existing languages have stood the test of time, haven't they?

Python Does Not Have GOTO

Consider UniBasic. Now I'm the first to expound on its advantages as a simple, clean way to express clear business intention. In the server world, clarity is king - especially when supporting the latest arbitrary business changes foisted on developers with few resources and fewer hours. But UniBasic has a problem - several problems in fact - that makes it a hard sell.

The first problem lies in the name. The only thing we can all agree about - whether we use UniBasic, UniVerse Basic, PICK Basic, DATA/BASIC, mvBasic or whatever - is that it has the word 'Basic' in it. That's not a selling point: you only have to look at the way Microsoft pitches its two .NET languages, VB.NET vs. C#, to see the problem. VB.NET (the BASIC one) is a language for hobbyists; C# is a language for professionals, even though what can be achieved is 99% the same. They pretty much compile to the same code.

Having a well known and loved language may well prove more attractive. That must be Rocket's hope. They know their customers need to attract new programmers.

Then, even for those who wish to use mvBASIC, there are problems of implementation when using the various

IT audits have you jumping through hoops?



PRC can help you meet your compliance requirements and make IT more agile and productive. No extra work, nothing to remember, nothing to fall through the cracks. Our software development lifecycle tool automatically prevents or detects change according to your criteria. You can deploy, rollback, test and report quickly, automatically and with confidence. Let PRC protect your company's valuable U2 data and software assets.



SJ+ Systems Associates • info@sjplus.com • <http://sjplus.com>



flavors. There's no unified standard: not only does the name differ between platforms, but, while I can write robust software for any one flavor, it is impossible to write anything, outside of the simplest CRUD style application, in a portable fashion. That is understandable, to an extent, between competing manufacturers. It makes no sense when multiple platforms are all owned by a single vendor. Writing for UniVerse, UniData and D3 as I do, the differences are a constant — *unnecessary* — overhead when it comes to releasing code. I wonder how hard it would really have been for Rocket to harmonize the language into a super-set that worked across at least the two U2 platforms, even if they held D3 aside. Implementing Python uniformly is an alternative to unifying mvBASIC.

Finally, we come to the question of inherited code. It is possible to write

beautiful, clean, modern, well structured, legible, fully tested, high quality, highly supportable and effective code in any flavor of mvBasic. But if mvBasic (or whatever you want to call it) has a bad reputation, it is primarily for a galling reason: It allows for the creation of some truly terrible code. Python gives us a reason to rewrite things instead of pouring even more badly-formed, ugly-looking hacks into the Charybdis of a legacy base.

The irony is of course, that in a very real sense, Python is the new BASIC. Beginners' All-purpose Symbolic Instruction Code – BASIC – was conceived as a language for teaching programming to non-technical students. Today, Python inhabits this same niche. The shelves at Amazon and other bookstores are full of works with titles such as Python for Complete Noobs Who Have Never Seen A Computer Let Alone Programmed One and Learn Python In Even Less Time Than That Other Book Promises. Raspberry Pi focuses on Python. Next year, England will become the first country in the world to mandate the teaching of programming to children from the age of five. Python is being touted as a possible choice for primary schools. Hackers are rumored to prefer Python for real-time attacks. Are we swapping one dirty hobby language for another?

What Is Python Good For?

Python certainly has many features that will be familiar to MultiValue programmers and to those who support MultiValue applications. It is compiled into intermediate byte-code and interpreted by a cross-platform virtual machine, just like mvBasic, with garbage collection and reference typing. Variables do not need to be declared before use because it has dynamic vari-

able typing. And more subtle things that speak well to MultiValue integration: strings can be single or double quoted, Boolean values are zero (false) or non-zero (true). That matches the way MultiValue databases work, unlike the more common 'true is -1 and 1 is false' other languages use.

Python is designed for rapid development and scripting. It is designed to produce highly legible code. Python aims squarely at the same people who historically would have been users of other non-technical languages (in the strict sense): business owners and rule makers. Its mantras, as set out by Tim Peters in *The Zen of Python (PEP_20)* <intl-spectrum.com/s/19z4F>, should strike a cord with MultiValue developers:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.

Python has very good documentation and tutorials. It runs on Windows, OSX and UNIX. And it's named after Monty Python's Flying Circus. What more could you want?

A MultiValue developer, wanting to try their hand at Python, will find as many differences as there are similarities. Some of these are subtle. For example, strings are immutable as they are in C#, so any changes to a string must create a new copy. Also, strings and array types, such as lists, are indexed from zero. And most programmers are, at least anecdotally, accustomed to the

SHORT-LINKS

What are those short links in the text of some articles?

As our authors share their knowledge and experience, they also point you to the Web for more info. Some of these URLs are long and cumbersome to use.

To make the URL easier to read and use, we are now using Short-Links to redirect you to the web page the author recommends.

The format (intl-spectrum.com/s9999) drops the "http://www." from the front.

Most modern browsers fill that in for you, or you can add it back in.

Comments or additional suggestions? Please drop us a note at editor@intl-spectrum.com

strangest feature of Python: the use of indentation for statement blocks. Never ever reformat a Python script!

There are other things which are, possibly, more welcome. Python has simple object orientation, with classes that bind together both functions and locally defined data. Python does it without the more formal aspects of other OO languages. There are no private methods, for example, and all methods are essentially class methods. Named arguments provide more flexibility in method calling, though they can be difficult to get your head around at first. Namespaces are used liberally to import functionality from other modules, leading to easy-to-follow and easy-to-extend scripts. But its biggest strength lies in the huge library of packages that can extend it. There are, as of the time of writing, nearly 50,000 packages on the Python Package Index <intlspectrum.com/s/19z81>. They cover everything from web communications to unit testing to image handling.

And it has a style guide: PEP 8 <intlspectrum.com/s/19z6t/>.

Ministry of Funny Walks

So Python is well-liked, extensible, and an easy step for existing developers versed in MultiValue development. Is that enough to extend the appeal of U2 to a new generation?

In my opinion Python is a safe bet. And, if the integration is done well, the same hooks could be used to accommodate other languages in the future. Rocket has worked hard to pack functionality into their databases. They have targeted the front end through an ever-broadening raft of interfacing technologies and APIs. However, Rocket has been more reticent in their

approach to the fundamental elements of the languages.

Their version of local functions, released last year, preferred legacy over innovation. While adding support for JSON style objects (UDOs) was a great step, it was marred by hiding it behind an ugly and obfuscating set of function calls rather than making it a first-class part of the language. They have never added the sort of object-oriented BASIC extensions found in OpenQM. Set against such a conservative background, in a conservative industry, any radical enhancement to our programming options should be applauded. The chance to bring in a new language, any new language, is a good thing.

The integration of Python with U2 is an interesting concept, and one that works both ways. Python developers (and therefore, Rocket customers hiring Python developers) get the ability to call into the U2 databases, execute subroutines and commands, and generally interact with the database in a sensible fashion. MultiValue developers get the ability to call out to Python from within their applications in a way which allows them use all the features of the extensive Python package library. Even if nobody is about to re-code their cherished applications in Python tomorrow, it is a good move that gives U2 sites a lot of benefits for free.

And maybe, for once, we can hang out with the cool kids. **IS**

BRIAN LEACH is a MultiValue developer, consultant, trainer and author, and a board member of the Rocket Software Users Group. You probably knew that. Find him at <http://www.brianleach.co.uk>.

WANT TO SEE A SPECIFIC TOPIC?

InternationalSpectrumislooking for writers, feedback, and topic ideas. We all have specific topics and issues that we need answers to find solutions for. Send us an E-mail with topics you would like to have covered in the magazine or on the website.

E-mail: nathan@intl-spectrum.com

WANT TO WRITE?

Expand your professional credentials, and provide us with an article.

Give us a rough and ugly outline, and we will help you refine it, proof it, and make it press ready. Or you can give us something polished, proofed, and press ready to publish.

Share your thoughts and expertise with over 10,000 fellow MultiValue developers and users.

E-mail: editor@intl-spectrum.com

NEED A MENTOR?

Mentors give developers the ability to ask industry experts for direction, code examples, and/or just ask them to see if something makes sense. Sometimes, all you need is a resource or example to start or complete a project.

Check with us to see who is available for mentoring, and how you can take advantage of it to save your business or company money.

E-mail: nathan@intl-spectrum.com

WANT TO BE A MENTOR?

We have many retired or semi-retired professionals out there that would love to share their knowledge of MultiValue development. If you are one of them, please contact us to see what mentoring is all about.

E-mail: nathan@intl-spectrum.com



International Spectrum Magazine

on Paper, Online, and Beyond

International Spectrum Magazine is moving beyond print and is now accessible on your favorite e-readers and mobile devices. By providing International Spectrum Magazine in the popular e-reader formats, subscribers are now able to take the magazine with them everywhere they go.

Let us help you keep your MultiValue skills up-to-date and efficient.

**Now on Kindle, Kindle Fire, Nook,
e-Readers, iPad, iPhone, and
Android Tablets!**



<http://www.intl-spectrum.com/magazine/>