

\$7.00 U.S.

INTERNATIONAL Spectrum®

THE MULTIVALUE  TECHNOLOGY MAGAZINE | MAY/JUNE 2016

Spectrum 2016 Conference Recap

Also in this Issue:

- Building a Modern Application (Part 3): Key Management
- Call the Parametrics!
- Is Agile Fragile?



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/>



INTERNATIONAL Spectrum

THE MULTIVALUE  TECHNOLOGY MAGAZINE

6 Spectrum 2016 Conference Recap

International Spectrum 2016: See what you missed at this year's four days of education, networking and fun. There were several Start-to-Finish tracks available to MultiValue developers. That's in addition to the hallway conversations, spur-of-the-moment consultations, and other interactions which make Spectrum a hit with the attendees.

FEATURES | MAY/JUNE 2016

10 **Building a Modern Line-Of-Business Application – Part 3** Key Management

The key to keys is the key to success. Creating record keys on-the-fly is common in software development, but a little planning can go a long way. This is part three of our article series exploring what it takes to build a Line-Of-Business application from scratch, using the tools and features found in modern software technologies. **BY**

NATHAN RECTOR

14 **Business Tech: Parametrics** Many of us are drowning in a sea of hundreds of programs. Parametrics allow us to severely reduce the total while still serving the same needs. If you believe that "code is a mortgage on your soul" read on.

BY CHARLES BAROUCH

18 **Specialization Versus Agile Development or, How to create high-performance teams, and love the generalists that can save your tail**

Once a shop has more than one programmer, the question of how to approach projects takes on a deeper dimension. As a solo artist, you have all the work, but also all the flexibility. Teams need at least a minimal structure. Agile has been touted as one of the best. Here's the counter-argument. **BY**

BENNETT BAROUCH

DEPARTMENTS

From the Inside page 4

From the Press Room page 13

Master Class in MultiValue page 16

Extending MultiValue page 23

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

What is the Most Valuable Asset in your company?

Is it the people? While good people are hard to find, and even harder to train, there is always someone that can fill that empty position.

Is it your company's inventory or property? While inventory and property cost money, they are always replaceable.

Is it your computer or office equipment? Much like your company's inventory, you can always replace it in case of damage or theft. Most companies have insurance specifically to handle this scenario.

What about your company's data? Is this an asset? YES. Companies often buy other companies for no other reason than to get their hands on the data. Your customer information – not just contact information, but also purchase history and other sales notes – that's one of your most valuable assets.

But sorry to say, while this kind of information is extremely valuable, and your company can't run their business without it, there is still one more asset that is widely overlooked: business applications.

The other day I brought this up with a colleague. He started to disagree, like many of you are likely doing right now, but then he gave it some thought. His first reaction was that the data is a company's most valuable asset - not just computer data, but also all those files stored in the deep, dark corners of your company that you actually have to get up out of your chair to access.

Data has a lot of value, but we all know that there's tons of data in

our company's databases that isn't tapped. Much of it is only tapped to answer one specific question. And then there's all that information in those file cabinets sitting the dark corner that you don't even use.

Your CEO always wants answers, and wants you to use the data you have to answer those questions. The thing is, the data you have to work with is limited by how and what your business application captured.

You hear about companies spending big bucks to implement a data mining or a BI (Business Intelligence) project and failing because they don't have the data to answer the questions. BI is only one part of the equation, and data mining is only as useful as the question that the user asks.

BI will not run your day-to-day business. Which brings us back to my original point, "Your company's business application is your biggest asset." Your business needs to run day-to-day to provide the mounds of data for the CEO and CFO to evaluate.

Most companies with high turnover positions have refined the business application to the point where training is minimal. Because the business application has such a level of refinement, these positions no longer have to have trained people in them to handle the day-to-day business. In some cases, the company can even eliminate certain positions altogether.

Please don't get me wrong! People are very important assets. The longer you can keep someone, the more stable your business will be. If a person leaves, it might hurt your

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., 3691 E. 102nd Ct., Thornton, CO 80229; Tel: 720/259-1356; Fax: 603/250-0664 E-Mail: request@intl-spectrum.com. Copyright 2016 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.

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

company for a few weeks or months, even if, overall you can always get someone to take over the empty position. The flip side is that the longer you keep someone, the more holes in your business application are developed.

Huh? If the person leaves, what happens to the company? Because the position was filled so long by someone who knew their job inside and out, your business application was not refined in that area. But now the person is gone.

Is that really a liability in the business application, or a liability caused by the person who was working in that position? They didn't point out the holes in application or create controls to help protect against this problem.

Thinking of the business application as an asset has helped many companies stay productive when key people leave, or - god forbid - go on vacation. I honestly don't know which is worse.

If the business can't run without a specific person, then that person is

no longer an asset to the company, they are a point of failure.

Like any other asset (property, people, or inventory), a business application can become a liability to the company. Your business applications need to be under the microscope so that they can show the company where their liabilities exist. They must be used to repair most of the problems that they highlight.

You, as a MultiValue business application developer, have the advantage of being able to modify your business applications quickly, efficiently, and without costing the company an arm and a leg. Most of the time, this means a return in profits. So the more a company is willing to invest in their MultiValue business applications, the more return they are likely to see.



NATHAN RECTOR

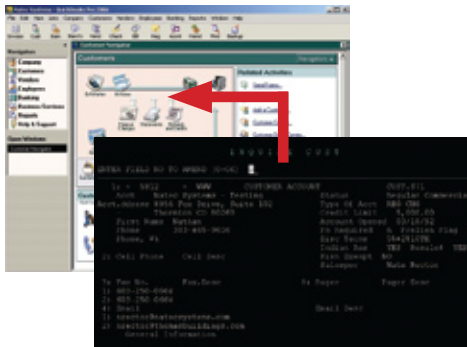
President

International Spectrum

nathan@intl-spectrum.com

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

Spectrum 2016 Conference Recap

If you are doing anything MultiValue, the 35th Annual International Spectrum Conference & Exhibition was the place to be for networking, solutions, and education. MultiValue professionals from North America and around the world gathered in Phoenix, AZ, from April 11th – April 14th, 2016, for a week of professional development and networking. Oh, and Google was there, taking about Big Table and how to make it work with MultiValue.

This year's attendees walked away with over thirty hours of professional development education in MultiValue database technologies and integration. And that was just from the sessions alone. This year's included topics for beginners as well as our advanced sessions. We covered MultiValue specific infor-

“We, as developers, can get caught in a rut and like to stay in our comfort zone. The conference brings together developers and vendors of different backgrounds and expertise. The Spectrum conference has given me new ideas and product solutions which I was able to implement in our business.”

mation as well as other topics of a more general nature.

To see a recap video for the 2016 conference, go to:

<http://www.intl-spectrum.com/resource/748/2016-recap-video.aspx>

During the Welcome Address on Monday, Nathan Rector, President of International Spectrum, talked about what was required for turn legacy applications into next generation applications. One of his key points, which set the tone for the conference, was the idea that building next generation applications was all about layering technologies together. Software development has evolved, not in leaps and bounds, but by building on the previous technology layers.

With that in mind, there were many topics on building integrated software, APIs, and design consideration that would take your business data from green screen applications into sharing data with GUI, Mobile, Web, Windows 10, Outlook, and other external applications. It was clear from these sessions that new application develop-

ment is about using the technologies that already exists, and enhancing it to do more.

Along with talking about how to extend existing MultiValue Applications, there was a lot of information for the new MultiValue developer and administrators. These sessions on the Fundamental of MultiValue database covered programming, dictionaries, and over-all system design.

There were several case studies done as well. One of the more popular ones was on “Hiring and Keeping Millennial Software Developers.” This session turned into a discussion that extended well past the session time.

On Monday’s lunch, we had a keynote session from Google. The topic was on cloud computing, and how it was no longer an “if” but a question of when and how. Seeing how Google is expanding Cloud Computing and how your MultiValue Application can take advantage of it, was very informative.

In addition to the keynote, there were several other sessions which discussed cloud and virtualization. With many businesses looking closely at whether

“This was my first time being able to attend the conference and I really enjoyed it. My stay at the Wigwam was pleasant and the weather was, of course, very nice in Phoenix. I really enjoyed the conference and all the good people that attended. I recently took a position with a company using D3 after working in Universe for over 20 years. Although I had a little exposure to the ten-line dictionaries I never really had to make use of them until now. Several people were very helpful explaining the details of f-correlatives to me.”

to buy new hardware or virtualize their existing systems, these sessions took on questions of speed, storage, and design consideration. There were even checklists for when and if you should plan

to virtualize into a private or public cloud.

Security and Encryption turned into a hot topic this year. With changes in credit cards and EMV/PCI compliance, identity theft, and crypto-viruses there was a lot of talk about how security your business information.

Following up the modern application design, mobile and web development was heavily talked about as well. These are no longer new technologies to the enterprise, but are turning into required technologies that companies need to implement.

On Tuesday and Wednesday, the attendees were able to talk to our sponsors about tools and products that would help their businesses. As always, we started with our Vendor Speed Dating, which is always a blast.

For those who do not know what Vendor Speed Dating is, all the sponsors get five minutes to talk to the attendees before the Sneak Preview party to inform them of things their companies can do for the attendees and the MultiValue marketplace in general.

You can listen to this year’s Vendor Speed Dating at:

<http://www.intl-spectrum.com/resource/750/2016-vendor-speeddating.aspx>

During Lunch and in the evening, attendees would gather in various places to Network with each other. One of the greatest advantages that the conference brings to the MultiValue community is its ability to bring together the novice and the expert to compare notes and talk about what issues they are having.





In addition to the non-scheduled networking time, we also included ten-minute appointment blocks where attendees could talk with our conference speakers about any topic they chose, outside of the sessions.

The conference wrapped up on Thursday with our traditional Closing Conference party. Attendees enjoyed one last time to talk, laugh, and network with each other. They also enjoyed the sun, food and drinks.

Most of the attendees took advantage of our post-conference downloads and session videos as well. The chance to revisit popular sessions and get access to downloads provided by the presenters and sponsors has proven very useful. If you didn't get a chance to attend this year, and would like to access to the session videos and downloads, you can still pay a fee and sign up here (free to attendees):

Thank You to our 2016 Sponsors

Ashwood Computer Company

BBWC LLC

DesignBais USA

Eagle Rock Information Systems

Entrinsik

HDWP

Kore Technologies

Ladybridge Systems Ltd

ONgroup Intl

Onsystem

PICK Programmers Shop

Rocket Software

SJ+ Systems Associates

Syngergetic Data

Zumasys, Inc.



”The International Spectrum was a great venue to learn some new technologies. I went on a web track, which was a beginning to end demonstration on how to build applications, not web sites, for devices like cell phones. The web track began with the most basic web, through PHP, style sheets and onto jQuery which makes the web pages into applications on the devices. These sessions are overviews and not really training, but the sources of the documentation are all given out at the sessions for further investigation. It all works with our MultiValue back-end.”

<http://www.intl-spectrum.com/conference/show/16/SessionDownloads.aspx>

Now that this year’s conference is over, it is time to start planning for the International Spectrum 2017 Conference. We’ll be back in Phoenix, AZ. See you there.

If you have a topic or session you would like to see at the Spectrum Conference, please let me know. I am always looking for new ideas and new presenters. **IS**

Modern Line-Of-Business Application

Part 3

BY NATHAN RECTOR

Last issue, we talked about logging and unique record IDs. There's still more to consider on that topic.

When planning different types of data structures, we should always start with record IDs (primary keys). There are framework considerations that should inform our decisions. It's a step which is often overlooked until it is too late. Even in relational databases, it is important to have a consistent logic for determining which fields should be used as the primary and secondary keys.

One-Database/One-Machine to Rule them All!

Aside from the "Lord of the Rings" reference, this is a common assumption made by many developers. We can get stuck in a mindset of "this will all run on one machine, in one location." While that is likely to be true of the development environment, it may not be true in production.

There's a tendency to start throwing together an application without considering the implications of what this mono-model represents. It is the simplest and fastest way to get the job

While VPN makes sense for individual users, it's a backwards answer here. If the bulk of the users are in one location, say a branch of a bank, putting the database local to the users makes more sense.

done, and that's all that matters to the programmer facing a killer deadline.

It does make record ID management much easier — in development. Ignoring the hard parts of the spec usually does make the job simpler. However, we'll find that the simple and fast approach has a lot of holes in it when we try to release it for production.

Some of the questions to consider before creating our Record ID management systems are:

1. Will we ever be selling the applications to others? If so, how do we want to manage the multi-tenancy?
2. Will the database ever be residing in more than one physical location?

If so, do we need to create a federated database?

3. Whether or not the database is split, what if the business has multiple locations? Do we need to consider multi-company/multi-branch records and, if so, how do we share information across companies or branches?
4. How do we handle file consolidation for central/corporate offices compared to branch offices or stores?

These questions bring into focus the drawbacks to the "one-database/one-server to rule them all" approach. Let's look at each of these questions individually.

Multi-Tenant Applications

Multi-tenancy is an architecture that allows a single application to service multiple customers and/or sites. This often comes up when building SaaS (Software-as-a-Service) applications, but the considerations we have to make for SaaS are the same as what we have to make when dealing with multi-company/multi-branch environments. Do we want to have each tenant (cus-

tomers) stored in the same database or do we want to create a separate database for each tenant?

Federated Database

A Federated Database is a structure that allows us to treat multiple autonomous databases as a single large database. This approach is often used in branch offices or brick and mortar store locations where most of the information can be handled independently from each other.

Most developers don't consider the advantages of federated databases because of the complexity it creates in their applications. It's a flawed idea to believe that "I'll just use a VPN to address the problem" is a reasonable alternative. While VPN makes sense for individual users, it's a backwards answer here. If the bulk of the users are in one location, say a branch of a bank, putting the database local to the users makes more sense.

The information still needs to be consolidated into a much larger corporate database for main office handling. Keeping IDs unique in the consolidation phase can be very complex. Much of that downside can be removed by choosing the right record ID management approach.

Federated databases also provide process isolation in addition to automatically mirroring and replicating systems without high priced replication, via software or hardware, being implemented. When we want to run a report for all the sales for the Colorado branch, then the report is run on Colorado's database and server, which does not affect the London branch, or Eastern Australia's.

The same goes for running large processes on centralized corporate data-

bases. Consolidated reporting does not affect any of the individual branches. It also provides fail-over backups.

Distributed Files:

Even if we don't want to get into a multi-tenant or federated database environment, we still need to consider the multi-company/branch question.

If we plan to grow our company, there will always be a need for the option to handle multi-branch and multi-company information. The multi-branch scenario is the more common situation for a line-of-business application.

How do we isolate the company or branch information from each other when we build a data-store? For example, if we have more than one physical store location, how do we handle each store's inventory information?

More than likely, we would want to share the general inventory information with all the stores, but each individual location would need to handle inventory running totals independently. The purchasing, receiving, bookkeeping, and P&L reports would need to be handled by-store as well as for the overall corporate information.

The easy answer is placing everything into one table, using a store or company prefix. The problem with this is that the table would quickly become millions of records, potentially causing scaling issue.

The alternative is to create a distributed table, where each store would hold its own information in a separate table. This allows better scaling when stores are added or closed. Distributed tables work well with federated databases or single databases. It does, however, add an extra level of complexity when

extracting and reporting consolidated information.

Software-Defined Record Key Management

Record-key management is best handled as a software-defined process rather than a hard-coded process that most developers tend to use. It fits the current trend of software-defined hardware, networking, and operating systems. There are advantages, but it adds a lot of up-front planning. The main advantage for software-defined record-key management is that we can choose one key-type for starters and then change it at a later date.

To do software-defined record-key management we need to make some decisions:

1. Single or Distributed Table – The configuration needs to define how to handle one-table-for-everything vs. distributed tables. This is about defining how item and table identities are handled.
2. Machine Independent Keys – If we plan to create federated databases, then all our keys *MUST* be machine independent. One way to do that is UUID V4. It can easily create machine independence, but the key size makes UUID hard to handle for data entry. The alternative is to create a machine, branch, or store prefix.
3. Sequential Keys – The sequential key will always be unique across distributed tables as long as we use the same seed. A common example of this is to have a control record called NextOrderNo that is used, and updated, by all stores. This only works in a single database environment. It will not be unique if we move into a federated database

4. Structured Keys – Structured keys, also referred to as derived keys, are always built around something unique. This is similar to using a machine/branch/store prefix, but it may not be as obvious. For example, all U.S. stores start with 5, the next two digits are the state code, and the next two are the tie-breaker for states with multiple stores. So, while the Wilmington, Delaware store might be designated #97, the structured keys might all start with 50104.

5. Key Reformatting – There should be some way to reformat the data when moving from a single table to a distributed and back again. This way the developer, or user, can change this information as needed. An example would be allowing every store to have an invoice #1, but prefixing the records as they are brought into the corporate system,

```
Call GET.KEY.MANAGEMENT ("CUSTOMER",KEY.MANAGEMENT, CUSTOMER.FILEVAR)
Call CREATE.KEY (KEY.MANAGEMENT,CUSTOMER.ID)
WRITE CUSTOMER.ITEM ON CUSTOMER.FILEVAR, CUSTOMER.ID
```

Fig. 1

and removing the prefixes when sending records back to the individual systems.

An example of a key management record would be:

Key: Table Name

<1> - Description of Table

<2> - File Structure: 1 - Single, 2 - Distributed, 3 - Federated

<3> - Key Structure: 1 - Machine Independent (UUID v4), 2 - Sequential, 3 - Prefixed Sequential, 4 - Structured (program driven)

<4> - Seed – Mainly used for Sequential keys.

<5> - Prefix Configuration

<6> - API/Program to call to generate Structured Key

Pseudo Code

See Figure 1

Conclusions

Software-defined key management may seem like overkill, but if you plan for this type of structure ahead of time, then many things become easier in the long run.

Stay tuned for part four. **IS**



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



taking multivalued ...
where it has never been before

- Close compatibility with most other multivalued environments
- Easy migration process
- Maintenance-free file system for ease of use
- High quality documentation
- QMClient API for development of GUI and web applications
- Low licensing cost
- AccuTerm bundled at no additional cost
- Many unique features

Ladybridge Systems Ltd
17b Coldstream Lane, Hardington, Northampton, NN4 6DB, England

Worldwide distributor: Zumasys, 9245 Reasearch Drive, Irvine CA 92618, USA
www.zumasys.com

www.openqm.com

FROM THE PRESS ROOM

Entrinsik Announces New Partnership with Atcore Systems to Provide Agile BI to SugarCRM Users

Entrinsik Inc has signed a partnership agreement with Atcore Systems, a leader in the Customer Relationship Management (CRM) design, implementation, and consulting arena, focusing within the SugarCRM platform. This agreement enables Atcore Systems to provide clients with the advanced reporting, real-time data analysis and dashboarding capabilities they need from an operational BI solution.

Informer enables organizations to perform ad-hoc reporting and analysis, blending data from multiple sources to create interactive reports and visualizations. Regardless of technical expertise, users can easily customize views, drill down into underlying data, dynamically group & sort, export & embed, all in real time, all on one screen.

Atcore Systems is the premier resource for designing and implementing CRM solutions to help businesses attract prospects, convert prospects to customers and retain the most profitable customers. Their experienced consultants are highly skilled in creating CRM and BI solutions to address an array of key busi-

ness pain points in sales and marketing, maximizing organizational productivity and competitiveness.

"We're really excited about being able to offer Entrinsik Informer to our clients," said Josh Sweeney CEO of Atcore Systems. "Their user friendly approach to blending data and creating visual reports aligns with our focus of making the client experience as seamless as possible."

"Atcore really stays in front of their clients' needs," says Tad Buck, Director of Informer Solutions. "As part of the partnership, Atcore will introduce Informer software into CRM solutions it deploys to help clients unlock the power of real-time reporting and advanced analytics regardless of where the data is stored." ■

Zumasys Announces Dates for jBASE 3.x and 4.x End of Service

Effective December 31 2016, the following products will move into End of Service status:

- jBASE 3.x on all platforms
- jBASE 4.x on all platforms
- jBASE 5.x on following platforms:
 - AIX 5.3 and earlier
 - HP/LUX
 - Red Hat ES 5 and earlier
 - Windows 2003 and

earlier

End of Service (EOS)

Zumasys will provide no further development on versions 3.x and 4.x and will no longer maintain jBASE versions 3.x and 4.x for problem replication. Licenses and general maintenance will continue to be available until End of Life. Customers on maintenance may contact Zumasys for assistance and obtain product upgrades during End of Service status.

End of Life (EOL)

jBASE versions 3.x and 4.x will move to End of Life status on June 1, 2017. The product version will then no longer be available for purchase. New licenses or changes in existing licenses will no longer be available. This includes changes to licenses for installation on new hardware, reducing or increasing user count, or any other change to existing license. ■

ONgroup Delivers FREE Production-capable MultiValue with MVON Express

ONgroup Intl now delivers their flagship MVON product with SQL Server Express. Both organizations and individuals can receive the product with the new \$0 annual license.

ONgroup has completed one MVON Express Tour with

the second underway. "We saw greater success than anticipated, with at least one Tour participant already deploying an application using MVON Express," said Charles Stauffer, President and CEO of ONgroup.

MVON SQL Server and MVON Oracle customers often use MVON to migrate their existing MultiValue applications to a SQL DBMS. Customers rely on ONgroup for smooth integrations because of the experience gained from 66 distinct product ports to date.

With MVON Express, a VAR can migrate an application or a single customer. A company can add a satellite system to their core MultiValue application, without swapping out their primary DBMS but with \$0 in annual license costs for the SQL Server Express and MVON MultiValue layers. Consultants and third parties can develop both internal applications and products using MVON Express.

For more information regarding ONgroup's ONware products and the opportunity to apply for the next MVON Express Tour or to try MVON SQL Server or MVON Oracle visit www.ongroup.com.

Product availability is FREE MVON Express working with FREE Microsoft SQL Server Express! ■

Parametrics

BY CHARLES BAROUCH

It's the early nineties. I've been tasked with fixing a truly terrible system at a small division of a big company. When I say terrible, I mean that the week-ending reports took eight days to run. Programs crashing. Different answers coming from the same data. I can still feel the pain just by thinking about it.

As I'm digging into the problems—weeks into the cycle of fix-test-fix—and in the shambles of a system before me, I spot a thing of beauty. The core of this behemoth was a small, tight, well-designed parametric system.

Call the Parametrics!

At that moment, I was transported back to a conversation I'd once had with my brother. Bennett said to me, "If you've ever written something and then put your hands behind your head and leaned back to marvel at how cool it is, you've written something too complicated."

Someone, early in the development of this system, wrote something marvelous. The disaster came as other people, with little or no understanding of the infrastructure, trod all over it. There are two lessons we can learn here. Either (A) don't ever do anything clever

...weeks into the cycle of fix-test-fix – and in the shambles of a system before me, I spot a thing of beauty.

or unusual, or (B) document things and invest in training your staff.

I've always favored plan B.

In the end, with two additional full-time programmers hired and an unmentionable amount of consulting money spent, the system worked... more or less. Some of what we did to fix it came down to stripping away the kludges and restoring the core design. The process was akin to rebuilding a classic car. What you ended up with wasn't what rolled off the assembly lines, but rather an homage to that original vehicle. Some parts the same, some parts newer and shinier, all of it covered with a coat of paint designed to make the results look uniform and consistent.

That marvelous parametric engine, in the hands of people who knew what it was, made the code versatile and responsive to business needs.

The Million Program March

In another job, we had two teams. Team Alpha was solving business problems quickly. Team Beta didn't show the same level of results. I was on Team Beta. More than that, I was the captain of Team Beta.

What was the difference between these two teams? Team Alpha wrote (mostly cut&pasted) an average of one new program per calendar day. Team Beta fixed the wreckage created by Team Alpha. We took fifteen versions of the same program and made one parametric version. We fixed `FOR I = 1 TO DCOUNT (REC, @AM)` so it didn't require the `DCOUNT` to recalculate for each of the thousand lines in the record. We killed off single letter variables like they were mortal enemies.

Why clean up? After all, computer hardware is getting cheap and programmers are relatively expensive. Just throw some Mflops at the problem, right? Thing is, as my friend Patrick once said, "Code is a mortgage on your soul." Three hundred and sixty-five new programs a year (give or take) made every change a nightmare to research. It made regression testing too time expensive. Without Team Beta, the system would have become the

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



biggest stumbling block to the future success of the company.

1K-Foot View

Parametrics are hard because you need to understand the mission. The thousand-foot view is required. Otherwise, you are randomly making some things parameters and not others. Let's look at a retail system, for example.

Prices fluctuate over time, but that's not a good place for parameters because it can be managed at the SKU or product level. However, sales affect categories, which means that building parameters which identify which items should be adjusted by which sales offers at which times, and so on. That's a good place for parameter logic. So, prices are pulled from INVENTORY records or a PRICE table, but sales are computed based on which settings are in play.

Selling men's shirts for Father's Day is a matter of adding a discount for each category which contains men's shirts. Set 30% for MENS-CASUAL, MENS-DRESS, MENS-SPORTY, and 0% for everything else. New holiday, new answers for the same parameters. No changes to the standard prices required. Nothing to change in the programming because the programs are parametric.

Here's the fun part. If we manage this by parameters and not by code, we can reap some sizzling secondary benefits. Your sales parameters can be used to tweak your restock logic so that you make sure that stores in your chain have the goods you are about to put on sale. Reporting can isolate sale-vs.-non-sale data because you have categories and time-frames. You can see which stores benefit from which sales. You can then target local events to different stores based on what brings in the crowd for each store.

```
read INV.REC from INVENTORY.FILE, INV.ID then
  PRICE = INV.REC<16>
end
```

Fig. 1

```
read INV.REC from INVENTORY.FILE, INV.ID then
  PRICE = INV.REC<16>
  locate(INV.REC<24>, SALES.CATEGORIES, 1, CPOS) then
    PRICE = int((100 - SALES.CATEGORIES<2, CPOS>) * PRICE / 100)
end
end
```

Fig. 2

People in midtown might buy differently than people in the suburbs. Or, maybe they don't. At least when it comes to your mix of products. Now you'll have the data needed to see which way your business works.

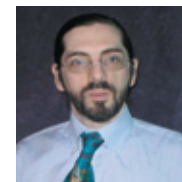
Code Example

The core of this idea is simple. Replace Figure 1 with figure 2.

What's happening on line 101? The SALES.CATEGORIES record has a list of category names (MENS-CASUAL, MENS-DRESS, MENS-SPORTY, WOMENS-CASUAL, WOMENS-DRESS, WOMENS-SPORTY) in attribute one. There's a corresponding discount (20, 20, 20, 0, 0, 0) in attribute two. When we want to give a different discount, we tweak the SALES.CATEGORIES record. For example, if we want to discount all "SPORTY" instead of all "MENS" we can set attribute two to "0, 0, 15, 0, 0, 15."

The Takeaway

You can't create a good parametric system without understanding the underlying business logic. Knowing what today's rules are is not enough. You have to have some sense of how they will morph over time. While including the subject-experts in your company is always a good idea, they supplement your understanding; they don't replace the need for you to know. **IS**



CHARLES BAROUCH is the CTO of HDWP, Inc. and the Publisher at HDWPbooks. You can read his writing in

International Spectrum, Theme-Thology, Novo Pulp, PerhelionSF, and the Interrogative series, which begins with Tiago and the Masterless.

IS.HASH.HMACSHA1

Generating HMAC-SHA1 Values with Generic MultiValue BASIC

With PHP, Python, and other languages, we can often find a complete answer – coded and ready – on the Internet. This article is part of an on-going effort to provide those sort of answers for MultiValue. Shared code is what makes any language easier to use.

The Message Authentication Code (MAC) is a widely used technique for performing message authentication. HMAC (short for "keyed-Hashing for Message Authentication") is a variation on the MAC algorithm. HMAC is currently an Internet draft that has been distributed by the Internet Engineering Task Force as Request For

Proposal (RFP) 2104. It has emerged as an Internet standard for a variety of applications. SHA1 (Secure Hash Algorithm 1) is the hash we've selected for this example of HMAC.

MAC and HMAC are procedures that allow communicating parties to verify that received messages are authentic. The two important aspects are: (1) verifying that the contents of the message have not been altered and, (2) that the source is authentic. There are several different hash codes that HMAC uses: MD5, SHA-1, and SHA-256 are the most common.

HMAC Function

The pseudocode in Figure 1 demonstrates how HMAC may be implemented. Block-size is 64 (bytes) when using one of the following hash functions: SHA-1, MD5, RIPEMD-128/160.

MultiValue BASIC Hashing

Most versions of MultiValue BASIC do not include built-in authentication hashing functions. This requires developers to create their own. It can be done in pure MultiValue BASIC, but you may see a performance hit, depending on how bit math is handled. For HMAC, the performance issues of

```
function hmac (key, message)
  if (length(key) > blocksize) then
    key = hash(key) // keys longer than blocksize are shortened
  end if
  if (length(key) < blocksize) then
    key = key || [0x00 * (blocksize - length(key))] // keys shorter than
blocksize are zero-padded (where || is concatenation)
  end if
  o_key_pad = [0x5c * blocksize] || key // Where blocksize is that of the
underlying hash function
  i_key_pad = [0x36 * blocksize] || key // Where || is exclusive or (XOR)
  return hash(o_key_pad || hash(i_key_pad || message)) // Where || is concatenation
end function
HMAC_SHA1("key", "The quick brown fox jumps over the lazy dog") =
de7c9b85b8b78aa6bc8a7a36f70a90701c9db4d9
```

Fig. 1

generic bit math does not seem to be a problem.

This IS.HMACSHA1 implementation <<http://www.intl-spectrum.com/r/754>> is dependent upon the IS.SHA1 hash subroutine. Be sure to

download the appropriate version of this as well.

To keep this program generic across as many version of MultiValue BASIC, I chose to use bit math to do the logical AND and OR, instead of any built-in

BITAND and BITOR found on some systems.

The code to replicate the Logical XOR needed in generating HMAC value was borrowed from Dave Meagher's excellent code found in the FOSS4MV/mvCrypt code on BitBucket:

<https://bitbucket.org/foss4mv/mvcrypt>

```
HMAC.KEY = "key"
HMAC.MSG = "The quick brown fox jumps over the lazy dog"
CALL IS.HASH.HMACSHA1 (HMAC.KEY, HMAC.MSG, HASH.VALUE)
*
TEST.VALUE = "de7c9b85b8b78aa6bc8a7a36f70a90701c9db4d9"
CRT HASH.VALUE : "=" : TEST.VALUE : "-"
IF (HASH.VALUE EQ TEST.VALUE) THEN
  CRT "Ok"
END ELSE
  CRT "Failed"
END
```

Fig. 2

```
de7c9b85b8b78aa6bc8a7a36f70a90701c9db4d9 = de7c9b85b8b78aa6bc8a7a36f70a90701c9db4d9 -
Ok
```

Fig. 3

THEME-THOLOGY: INVASION

Voices by Lisa A. Kramer
I Was a Teenage Alien by LJ Cohen
Singularity by Jeremiah Lewis
Not Like Us by Mike Reeves-McMillan
That Kind by Charles Barouch
Yellow by Bill Ries-Knight
An Invasion of Ideas by Jeremy Lichtman
Famine, with Fries by Jefferson Smith
The Several Monsters of Sainte-Sara-la-Noire by Michael Williams
Going Viral by Rachel Desilets
Dead Planet Scrolls by Timothy Hurley
Red Vapor by Michaela Susanne
The Worms Crawl In by Michelle Mogil
Nano Nation by CM Stewart
The Woods, The Cellar, and Cover art by Aaron Wood
All other interior art by Juan Ochoa

WWW.THEME-THOLOGY.COM



Specialization Versus Agile Development

or, How to create high-performance teams, and love the generalists that can save your tail

BY BENNETT BAROUCH

One of the less talked about goals of some Agile approaches is to eliminate distinctions between engineers and make each team member a solid generalist. Ideally, anyone on the team should be able to take on the current highest priority task in the backlog, regardless of the knowledge and skill level required to complete it.

Benefits:

- Everyone writes better product code when they understand thorough testing. Everyone writes better tests when they understand the numerous pathways that exist in the code, and when testing is understood as a multi-dimensional and demanding discipline of its own.
- When they understand the overall system, its various components, and the end user's objectives, knowledge, and work-flow, engineers can make richer contributions to the product than when they just know "their part."
- People write better software on one side of an API when they also have experience on the other side.

Use cross-training in support of a culture in which people with significantly different types of brains are likely to develop genuine appreciation of each other's work, and of each other.

- Not having to share specialists across teams reduces scheduling dependencies and delays, and allows each team to be truly self-sufficient, or whole. Additionally, this leads to a stronger sense of responsibility, to more complete team bonding, and to more team learning and efficiency.
- The closer we get to everyone being able to do all the different kinds of work there are, the higher our staff utilization, the more robust we are, and the more quickly we get high quality product to market.

It's all good. Except for the fantasy part. That is, it's a great idea, but since I have no expectation of attaining it, I

want to talk about what we can realistically achieve.

First, why we won't get there: We have just described a significant subset of what it means to be a — cue the blaring trumpets — *Great Software Architect* which is not where everyone in the industry is going, so we know empirically that this goal will not be reached. Understanding why is this will help us guide our actions.

One reason is that not everyone wants to get there. To list just a few of many possible reasons, some want to only be responsible for the work they do with their own hands, some really like being a specialist, and some want to go into management, or product management. But an all-generalist team would be a poor plan even if everyone on your staff did want to become a great architect.

The kind of thinking that runs through single-threaded, interrupt driven, dynamically typed, front-end JavaScript with brilliance might not be adept at multi-threaded, stateful and process-oriented, type-safe, back-end Java programming. The brain that intuitively perceives end-user goals, relevant metaphors, order of user operations, and

powerful presentation of results is unlikely to also do outstanding work in performance, scalability, extensibility, and security. A person skilled at building great general-purpose back-end business logic and library code is not automatically great at creating applications that handle even basic usability considerations.

Our vocational experience contributes significantly to what kind of thinking we do best. That might seem to imply that broader experience will automatically produce a broader range of thinking styles, but the data do not support this hypothesis. Whether due to biological brain differences, early life influences, acquired personal preferences, or declining mental flexibility, most painters do not bounce back and forth between impressionism and art deco, and most database people are not the ones you want creating your UI, nor do they want to do it.

When the rare someone is great at some key aspects of software product development and pretty good at several others, they are strong generalists, and they have the potential to become great architects. For combined reasons of different interests, different brains, and different accidents of experience, this is a small subset of those staffing our engineering efforts. Thus, for most products, and certainly most enterprise products, we should plan on hiring several people, with different skills, and sustaining a culture that helps people with different kinds of brains and experiences to work together effectively. Oh, damn. This is about social facilitation, not engineering.

It's about people with all their complexities, not inanimate things we can shuffle around without consequence. Several studies, including one done recently at Google, have found that the best-performing teams are not those

with the most relevant expertise(!), nor those that work the longest hours, nor those with strong leaders. They are those in which unplanned communication is most fully shared – does everyone in a team meeting speak a similar amount of time, and to similarly engaged listeners.

These studies conclude that we need to talk with each other with a lot of social and vocational safety – it has to be safe to not know, to be unsure, to be wrong, to investigate and experiment, and to disappoint, in order to achieve greatness. Most engineers and most companies simply won't tolerate this. Most are achieving less than great results. Hmm.

The Google study also says team members need to be able to depend on each other, operate with clear goals, roles, and plans, and care about the meaning and impact of their work. In too many companies, “dependable” means you do what I ask, when I ask, regardless of the infeasibility, or cost to your family life, *clear goals, roles, and plans* comes down to the same thing, and *meaningful work with impact* means whatever it is that this company does — you're lucky to work here, so suck it up.

By contrast, the best-performing teams are dependable because they communicate in a safe environment in which they feel so respected, valued, and supported by other people's efforts that they don't want to let anyone down — that's a positive social bond, not the pressure and fear that are the driving forces in so many companies.

High-performing teams have clear goals, roles, and plans, because they create them together, and as the work progresses and these things all inevitably flex or change outright, safe, high-

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.

bandwidth communication keeps everyone aligned.

People on these teams might find meaning and impact in their work because of the company's mission, but what is likely to matter much more is whether they enjoy the problem-solving challenges in the work they do, feel pride in their increasing accomplishments, and find their relationships with co-workers and their boss meaningful. How many times have we heard that people don't quit jobs, they quit bosses and work environments where they don't feel that they are valued, or where they feel they have little left to learn?

This was a longer digression than I would have liked, but perhaps it helps us see that if we want to have a high-performing team, our goal should not be to make engineers into the interchangeable inanimate objects that they will not become. Rather our goal is:

Use cross-training in support of a culture in which people with significantly different types of brains are likely to develop genuine appreciation of each other's work, and of each other.

Our cross-training programs, when they have existed, have had the wrong goal, and have largely failed. The goal — my goal anyway — is not to get the UI guy to be a drop-in replacement for the database gal. That's well, um, stupid. The goal is for the UI guy to understand just enough about the engineering strengths and constraints of database programming to ask better questions, make better suggestions, ask for and provide better interfaces, and do better work *as a UI guy*.

How do we hit this more realistic target? The battle is largely won just by communicating that this believable

result is the goal. In practical terms, use pair-programming to cross-train the UI guy to do just enough database work to have a feel for what is going on down there. Do just enough for the database gal to have an understanding of UI and UX challenges. Ensure this leads to communication between these layers becoming more efficient and less error-prone, and communication between these people more informed, respectful, trusting, and synergistic. That's teamwork, that's tactically and strategically valuable, and that's achievable. It's also the larger part of what managers should be measured on, as everything bottom-line flows from this, and everything in "soft skills" is embodied in this.

Let's begin by having our architects and technical leads describe the system neither in superficial "marketecture" terms, nor in details only specialists will understand. Let them use cross-discipline analogies and note cross-discipline differences. Charge them with the goal of making each specialization interesting and respect-worthy instead of *strange and not my thing or not real engineering*.

Benefits:

- Most staff members should perform better in their existing jobs
- Some will become able to work skillfully in multiple areas
- Some will end up ranking among your most valuable employees — only because you gave them this chance to grow

Do not expect cross-training (and certainly not the absence of cross-training) to result in everyone becoming interchangeable at management's convenience.

Note one exception to this general line of reasoning:

Go for deep cross-training between traditional testers and traditional product developers. If this does not yield better thought-out code with fewer defects, and better tests within a few months, management needs to figure out who needs what additional coaching, or perhaps who needs a new job!

That's it. Modest goal, modest investment, immodestly better teamwork and resulting product.

See how I snuck that claim in at the end there with no data to support it? I don't have any data other than my own casual observation. If this seems like a lot of hooey to you, ignore it. If this has seemed pretty sensible to you, take this approach and get people talking about the results that you obtain. Make it better. Share your experience more broadly.

You might suspect me of not believing in the generalist brain — so let me be explicit about that. A generalist mind is itself a kind of specialization. Generalist minds gravitate to the thematic, to patterns that often go undetected amidst noisy details, to semi-universals and how to know which to apply in a given case.

It would seem generalists are where eventual *great architects* are most likely to come from. After all, great architects are not people who are only good at one or two things, or at taking only one or two perspectives into account, although many such people are given the Architect job title. Great architects see patterns across diverse domains.

Require good performance but don't require the generalist mind to be your best performer in every category — perhaps not even any one category.

Judge your generalists by their work product results, not by the difficulty some generalists have relaying details when questioned. Demanding details of a mind that constantly abstracts patterns seems reasonable from the outside, but can be paralyzing to some generalists.

Learn to work with them so they want to keep working with you. Honor his or her interest in moving through a range of jobs, synthesizing the knowledge and judgment required to gradually become one of the most valuable people you can have on your team.

Note too that a strong generalist who never makes it to the great architect level can still be one of the most valuable people on your team — often able to assist in diverse situations, and frankly, to cover your ass in innumerable emergencies. Some generalists are

brilliant decision makers, and inspirational communicators, even when they can't articulate the details that have actually led them to their conclusions.

I recommend the following advice for all employees, but most especially for generalists:

Train people well enough so they can leave; treat them well enough so they don't want to.

Richard Branson

Since I cast this in opposition to a common Agile goal, I want to give a small sidebar: I am strongly in favor of Agile in general, and scrum in particular. I most emphatically recommend that people stop customizing its project management model, and then complaining that “Agile doesn't work” when they have declined to change their own most counter-productive

practices. But the idea that all players on the team should be interchangeable is oddly surrealistic and out of place in a line of thinking otherwise grounded in empirical evidence.

Different kinds of work are best performed by people with different kinds of minds and different personal goals — these are real differences, not arbitrary divisions of labor. We should move in the direction of cross-team awareness (not de-specialization), in expectation of improved individual and team performance. **IS**

BENNETT BAROUCH has over 30 years of industry experience spanning design automation for integrated circuits deployed in satellites, financial portfolio software, high transaction volume and big data systems, information management, secure on-site and mobile networking, and IT operations software. His customers run from individual retail consumers to the largest companies in the world.

Groundbreaking work under Bennett's leadership produced a virtual assistant that could understand 20 million English phrases and respond with a wide array of information and with complete computer-telephony integration. This work was made part of the permanent collection of the Smithsonian Institution, for Outstanding Achievement in Information Technology, 14 years before Apple released Siri and became the basis of the OnStar virtual assistant found in GM automobiles. Bennett has been certified in ITIL, and as a Scrum Master and as a Scrum Product Manager.

can
MultiValue can't do that.

Change your thinking.


Modern MultiValue, LLC
Modern Solutions for Modern Applications
info@ModernMultiValue.com
www.ModernMultiValue.com
+1.225.341.1778

MARKETPLACE

ACCOUNTING

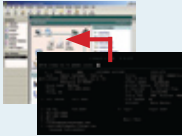
Natec Systems

www.natecsystems.com | nrector@natecsystems.com




QuickBooks API for the MultiValue Database

- Read/Write Directly to Quickbooks Databases
- mvQB API is Designed for the MultiValue Program to Use
- 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

COMPLIANCE

SJ+ Systems Associates

www.sjplus.com | sjoslyn@sjplus.com

CONSULTING

Drexel Management Service

www.drexelmgmt.com | dconboy@drexelmgmt.com

Execu-Sys, LTD

www.eslny.com | mh@eslny.com

HDWP

www.HDWP.com | results@HDWP.com

Modern MultiValue, LLC

www.ModernMultiValue.com | info@ModernMultiValue.com

PICK Programmers Shop

www.pickprogram.com | brian@pickprogram.com

Precision Solutions

www.precisiononline.com | Kevin@PrecisOnline.com

DATABASE

Ladybridge Systems Ltd

www.ladybridge.com | sales@Ladybridge.com

REPORTING

Brian Leach Consulting, LTD

www.brianleach.co.uk | brian@brianleach.co.uk

TERMINAL EMULATOR

Zumasys

<http://www.zumasys.com/products/accuterm/>



AccuTerm® software, the leader in terminal emulation, allows you to access your MultiValue application—whether it is on-premises or in the cloud—from any Windows device.

www.zumasys.com/accuterm
818-951-1891



LETTERS TO THE EDITOR

Have an opinion on an article: Agree, disagree, or enhancement to an article from a previous issue? International Spectrum and our authors are interested in hearing from you!

E-mail: editor@intl-spectrum.com

WANT TO SEE A SPECIFIC TOPIC?

International Spectrum is looking 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

CALLER and GET .CMD

BY CHARLES BAROUCH

CALLER is a TCL command which calls a subroutine. It is useful for testing. Full explanation at <<http://www.intl-spectrum.com/r/757>>

Example: CALLER DOZ.CONVERT 144 (P)

```
000 CALLER
001 * From intl-spectrum.com/xmv/caller
002 call GET.CMD(CMD,OPT)
003 CSIZE = dcount(CMD,@FM)
004 dim PARAMS(100)
005 for PCNT = 1 to 100
006   PARAMS(PCNT) = CMD<PCNT>
007 next PCNT
008 begin case
009   case CSIZE = 4
010     call @CMD<2>(PARAMS(4))
011     print PARAMS(4)
012   case CSIZE = 5
013     call @CMD<2>(PARAMS(4),PARAMS(5))
014     print PARAMS(4),PARAMS(5)
...999 end case
```

GET.CMD is a subroutine command which gets all of the command line values used. Full explanation at <<http://www.intl-spectrum.com/r/757>>

```
000 GET.CMD
001 * From intl-spectrum.com/xmv/get.cmd
002 GET.CMD(CMD,OPT)
003 CMD = SENTENCE()
004 OPT = ''
005 if (index(CMD,'(',1) then
006   OPT = oconv(CMD,'g(1');* This is not foolproof
007   CMD = oconv(CMD,'g0(1')
008 end
009 convert ' ' to @AM in CMD
010 return
```

This is an occasional mini-column for showing off quick bits of code which extend TCL, mvBASIC, Proc, or other MultiValue features and functions. Want to submit your own? E-mail editor@intl-spectrum.com with your XMV.