Converting
SQL TABLES
to
MultiValue Files

Customer
Customer_id
Firstname
Lastname
Address
Postal_code
Age
Gender
Email
Order
Invoice

Product
Product_id
Product_name
Amount
Price
Description
Image
Date
Status
Statistic

Invoice
Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

Order
Order_id
Total
Product_id
Customer_id
Date_time
Remark

Also in This Issue:

- **DIY SSL – Self Signed Certificates**
- **The Value of MultiValue**
- **Philosophy of Indexing**

INTERNATIONAL SPECTRUM

SAVE THE DATE!

MULTIVALUE CONFERENCE

2019

APRIL 8 – 11, 2019

**THE WIGWAM
LITCHFIELD PARK, ARIZONA**

REGISTER AT:
www.intl-spectrum.com

THE PREMIER MULTIVALUE USER EVENT

INTERNATIONAL SPECTRUM

THE MULTIVALUE TECHNOLOGY MAGAZINE

CONTENTS

From *the* Inside

Gone are the days when your IT team was the only ones in the office who knew how to use technology. This change in tech aptitude has changed the approach to enterprise software. While that space has historically been quite slow to evolve, 2019 is set to bring about a shake-up to the status quo. Most of our companies seem to be struggling with where to go and what to do to keep relevant in the changing business world.

I've compiled a few trends that will likely continue to see getting more significant in 2019.

#1 Smart Enterprise Software

Your business software already has tons of "smarts" built into them, but your enterprise has taken much of it for granted. Many times changing columnar reports into dashboards or adding simple push notifications will turn a bland enterprise system into something that acts like a Smart Enterprise System.

The next step to producing smart systems is to introduce some form of AI or Machine Learning tools into our system. How and why is not always straight forward, but with many systems, using the new predictive inventory or sales models is a good starting point. We had a session on predictive inventory at the most recent International Spectrum Conference.

#2 User Experience

I've discussed User Experience (UX) for many years as a form modernization. The point of UX is to provide more efficient and productive output and input to the enterprise system. This results in a better customer experience because employees can quickly and efficiently do their work. Unfortunately, we seem to focus on GUI-as-UI and ignore the importance of integration. Good UX is end-to-end.

UX has waned a little over the last few years due to the complexity and cost of implementing it, but I've seen a renewed focus on UX in enterprise software. And the focus on UX is finally changing from "just build a GUI" to addressing the back-end issues.

The many users understand technology — and many more think they do — and that leads to DYI (Do It Yourself) happening outside of the IT department. Excel is a classic example of this. It is a two-edged sword. User-ownership of data is a good thing but inexperienced people guessing through data-relationships can lead to bad things.

Your job has become more social. We can't just deliver results, we need to talk to the users and bring them inside the tent. Chatting and communication (Microsoft Teams, Slack, many other tools) provide another way to improve UX without losing the tried and true Enterprise system.

twitter.com/intlspectrum

intl-spectrum.com/facebook

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.

#3 Automation

More and more enterprises are starting to recognize that automation can have a positive impact on user experience.

Enterprise Software should aim to automate the entire business process. Instead, we often create islands of automation.

These were typical in the past. Humans were the only bridge over the gaps between different bits and pieces of information. Traditionally, customers would talk to employees who could capture necessary information on internal systems.

But now we want customers to interact directly with the business via software, websites, mobile apps, and digital assistants. This makes processes quick and simple for customers who want to get on with their lives. It also means that we have to make sure that nothing is lost in the transition.

#4 Cloud and Hybrid-Cloud

Cloud has always been described as a platform for the next generation of business. It is here to stay, and many companies are finding increased benefits. Many are not be interested in placing their entire enterprise in a Public Cloud Datacenter.

Parts of the enterprise will go to the Cloud, but other parts like real-time and business critical systems will stay on-premise. This Hybrid-Cloud approach will become more and more common. The need to have these systems communicate with each will add an additional complexity to your already integrated systems, but will open doors to many other features that aren't currently available to on-premise only enterprise systems.

Modern enterprise solutions and apps need to be as good as the apps people already use and love, which is quite a departure from the average enterprise system in the wild today. **IS**

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

Migrating
NorthWind Database
to a MultiValue Database

BY NATHAN RECTOR

The battle between traditional database (RDBMS) and MultiValue databases have raged for years, and will likely continue for the foreseeable future. As NoSQL-style databases are starting to become more accepted by CEOs and CIOs for enterprise-level datastores, the MultiValue model is being looked at very closely again. The old, unfair, stigmas are gone.

RDBMS were never really designed for high volume CRUD (Create, Read, Update, and Delete) business transactions, where the MultiValue Database was designed specifically to handle and store business data. Other NoSQL databases are gaining ground, but they don't have the enterprise level locking, CRUD management, transaction boundaries, and general OLAP transaction ability that MultiValue Systems have implemented over the last 30 years of their existence.

> *One of the benefits of turning the OrderDetail data into a group of MultiValue fields inside the Order File is "Referential Integrity"*

This flexibility and ease which is core to the MultiValue datastore can be confusing to outside developers. Time working in other models creates expectations that don't match the MultiValue experience. They find themselves trying to solve problems which don't exist in our systems. And, without properly understanding how to construct data models effectively, they won't get all the benefits.

To demonstrate the differences, I decided to use Microsoft's sample database, NorthWinds, as an example. Why? Because most developers will have run across it while learning how to interface with an RDBMS. Giving them a familiar point of reference will help them compare and contrast the techniques needed to optimize in each environment.

Good programmers don't just know code, of course. As with any development project, you have to understand the business — the what, why and how — which informs the need of the software. If you don't, your project will fail. For those not familiar with North-Winds data, it is a fictitious company called NorthWind Traders that creates and processes orders. This collection of data has been around for years and has been replaced with more complex data models like AdventureWorks but it stil stands as a clean and clear example.

NorthWinds Business Model

Figure 1 shows the schema of the NorthWinds database as found in Microsoft SQL server. Let's assume the company creates new orders ev-

Figure 1

ery day. The number of orders created could be in the tens of thousands per hourly. It all depends on how many users are entering order. With this assumption of high volume, we can further assume the Orders entity is write intensive. Reads, on the other hand, will not be as voluminous as writes. In other words, Orders is, in MV-speak, a transaction file.

Next, let's consider the entities of Customers, Employees, Products, Shippers, and Suppliers. These entities will be created, updated, or removed at a low frequency. These are, in MV-speak Master Files. Product reads will be particularly high against every Order. An order usually has only one Shipper or Supplier but many Products.

Building the Database in MultiValue

Now that we know the business expectations, let create the basic files and dictionaries that match the SQL Structure above. I'll lay the structure out using standard MultiValue LIST-DICT syntax [*Figure 2*].

File: Order [*Figure 3*]

As you can see, this is pretty straightforward. You can do similar conversions with other tables, but there is something important we have to address: OrderDetail. This table isn't a primary table like most of the tables in the Northwinds database.

It is what SQL administrators like to call a subtable, support table, or detail table. Basically, it is a table that can't live on its own, and requires the Order table to

```
/* Table: orders */
CREATE TABLE orders (
  id               INT NOT NULL,
  employee_id      INT ,
  customer_id      INT ,
  order_date       DATETIME ,
  shipped_date     DATETIME ,
  shipper_id       INT ,
  ship_name        VARCHAR(50) ,
  ship_address     VARCHAR(250) ,
  ship_city        VARCHAR(50) ,
  ship_state_province  VARCHAR(50) ,
  ship_zip_postal_code VARCHAR(50) ,
  ship_country_region  VARCHAR(50) ,
  shipping_fee     DECIMAL(19,4) NULL DEFAULT '0.0000',
  taxes            DECIMAL(19,4) NULL DEFAULT '0.0000',
  payment_type     VARCHAR(50) ,
  paid_date        DATETIME ,
  notes            VARCHAR(250) ,
  tax_rate         DOUBLE NULL DEFAULT '0',
  tax_status_id    TINYINT ,
  status_id        TINYINT NULL DEFAULT '0',
  PRIMARY KEY (id)
);


/* Table: order_details */
CREATE TABLE order_details (
  order_id             INT NOT NULL,
  product_id           INT ,
  quantity             DECIMAL(18,4) NOT NULL DEFAULT '0.0000',
  unit_price           DECIMAL(19,4) NULL DEFAULT '0.0000',
  discount             DOUBLE NOT NULL DEFAULT '0',
  order_detail_status VARCHAR(25),
  date_allocated       DATETIME ,
  PRIMARY KEY (order_id, product_id)
);
```

Figure 2

help provide additional context to the data found in OrderDetail table.

In SQL, you would have to do a JOIN to get the OrderDetail information along with the Order information [*Figure 4*].

Since the OrderDetail information is not likely to be accessed independently from the Orders file, now would be a good time to take advantage of the MultiValue database model to store the OrderDetail data.

There are a few ways to handle this in the MultiValue database:

1. Create it exactly the same way, we can do anything architecturally that SQL can.

2. Create a group of MultiValue fields inside the Orders file

3. Create a Group of MultiValue fields outside the Orders File

There are several design arguments for and against using both ways of doing this. This is where MultiValue developers have options denied to SQL developers. We get to choose.

OrderDetail - MultiValue Fields Inside Orders

One of the benefits of turning the OrderDetail data into a group of MultiValue fields inside the Order File is "Referential Integrity".

Another benefit has to do with data retrieval and update optimization. If the OrderDetail is included with the Order data, then one IO read will pull all the data that is associated with the Or-

Name	AMC	Description	Conversion	Correlative	Justify	Display Length
ORDER_ID	0	Id			R	10
CUSTOMER_ID	1	CustomerID			R	10
ORDER_DATE	2	Order Date	D4/		R	10
SHIPPED_DATE	3	Shipped Date	D4/		R	10
SHIPPER_ID	4	ShipperID			R	10
SHIP_NAME	5	Ship To Name			L	50
SHIP_ADDRESS	6	Ship To Address			L	250
SHIP_CITY	7	Ship To City			L	50
SHIP_STATE_PROVINCE	8	Ship To Region			L	50
SHIP_POSTAL_CODE	9	Ship To Postal Code			L	50
SHIP_COUNTRY_REGION	10	Ship To Country			L	50
SHIPPING_FEE	11	Shipping Fee	MR4		R	10
TAXES	12	Taxes	MR4		R	10
PAYMENT_TYPE	13	Payment Type			L	50
PAID_DATE	14	Paid Date	D4/		R	10
NOTES	15	Notes			L	250
TAX_RATE	16	Tax Rate	MR4		R	6
TAX_STATUS_ID	17	Tax Status			R	4
STATUS_ID	18	StatusId			R	4

Figure 3

```
SELECT Orders.orders.id, Orders.customer_id, Orders.Order_date, OrderDetails.Quantity,
OrderDetails.unit_price
FROM OrderDetials
WHERE Orders.Id = "1234"
INNER JOIN Orders ON OrderDetail.OrderID = Orders.ID
```

Figure 4

der. In turn, one IO write will update all the data associated with the Order.

This differs from the SQL model which would require an IO read/write for every OrderDetail record associated with the Order. No matter how fast your hardware is, one write will be faster than fifty writes on that same hardware.

File: ORDER [*Figure 5*].

When you are ready to see the information you would enter a command as seen in Figure 6.

The drawback to this approach starts to show up when you have Orders that contain tens of thousands of lines of OrderDetail. While a single Multi-Value record can easily handle tens of thousands of MultiValue, it can slow down filtering and selections.

For example, if the user needed to report on all the Orders that contain part ABC, then the following select would look something like Figure 7.

This looks straight forward, but it would cause the selection to read each record in the Orders file and parse the multi-valued field in memory to find the part number. The selection process then returns the indexed position – the multi-value position – of the part number.

Once the select is complete, the list statement would have to parse each of the associated multi-valued field, QTY and UNIT_PRICE, to return the particular value found at that relative position.

While parsing and processing multi-valued fields that are only thousands of values big is very efficient, when you get into the ten thousand, the overhead of doing an additional IO read may be better.

In later articles, I will look advantages and disadvantages of the keeping the OrderDetail as a single file, as well as, grouping OrderDetail as multi-value, but keeping them in a separate file. **IS**

Name	AMC	Description	Conversion	Correlative	Justify	Display Length
PRODUCT_ID	19.m	Product ID			R	10
QTY	20.m	Qty			R	10
UNIT_PRICE	21.M	Unit Price	MR4,$		R	10
DISCOUNT	22.M	Discount	MR2,$		R	10
DATE_ALLOCATED	23.M	Date Allocated	D4/		R	10

Figure 5

```
LIST ORDERS "1234" ORDER_ID CUSTOMER_ID ORDER_DATE QTY UNIT_PRICE
```
Figure 6

```
LIST ORDERS BY-EXP PRODUCT_ID = "ABC" ORDER_ID CUSTOMER_ID ORDER_DATE QTY UNIT_PRICE
```
Figure 7

BY MICHAEL BYRNE, PRINCIPAL SOFTWARE ENGINEER
ROCKET U2

Self-Signed SSL Certificates for REST Development

Security can be a big concern when you are developing APIs that are exposed to the outside world. Best practice dictates that you should use SSL/TLS to secure your endpoints by encrypting the traffic if you are dealing with any sensitive information. SSL (Secure Sockets Layer) is a global standard security technology that encrypts the data being transmitted and authenticates the identity of the website. TLS (Transport Layer Security) is actually the newer replacement of SSL, but most people are more familiar with the historic term SSL so you may see that term used even though it actually may be TLS as the protocol. Standard HTTP traffic is unsecured and sent in clear text, but HTTPS adds a layer of security by using an SSL/TLS protocol connection to encrypt the data being sent.

SSL Certificate Overview

To ensure that the server is actually what it claims to be, an SSL certificate issued by a trusted third party is used to validate the claim. This third party is called a certificate authority (CA) and there are a number of major recognized companies such as Verisign, GeoTrust, and Comodo that provide

> *In production, you should obtain and install a certificate from one of the major CAs.*

this service. These companies will verify the identity of the company/person requesting the certificate, and in turn, when they sign your SSL certificate with their root CA certificate your browser will trust that certificate.

In production, you should obtain and install a certificate from one of the major CAs. Most of these major root CA certificates are included in your local trust store by default, so if an SSL certificate is signed by one of these, your browser will accept its authenticity and continue normal operation with the HTTPS connection. However, there are sometimes cases where you don't necessarily need the recognition and cost of an official certificate. For example, if you are working in a development environment and you just need to make sure that your application works with SSL/TLS, you can "self-sign" your own certificates.

Become Your Own Certificate Authority

You could create a specific certificate for each required server, but then you have to add each individual SSL certificate into your trust store every time you create a new one. Another option is to become your own CA, and once you install that root certificate, any other SSL certificate you sign with that one will be trusted.

Programs You Need

There are a couple of OS utilities you will need to generate all the necessary keys and certificates. The latest UniData and UniVerse releases also include some Basic functions that will help you create certificates, but in this post we are focusing on the underlying technology so you can understand what is happening. You can also read this blog post about the Certificate Management Tool, which hides some of this complexity from you with a set of scripts to walk you through some key generation tasks.

- **openssl** – Utility from OpenSSL toolkit used for key management. This is provided in the U2 bin directory.

- **keytool** – Java keytool is a key and certificate management utility. Should be in JRE bin directory.

Create Your Root CA Certificate

You will be asked a set of questions to define this certificate. This is the information that will be seen when you view the details of this certificate. Because I will be using this certificate to sign other SSL certificates, I will just use my name as the Common Name so that I can recognize it when looking at my list of trusted certificates.

Note: I am using the bash shell on Windows for this demo

```
# Create your private key
openssl genpkey -algorithm RSA -out myprivate.
key -pkeyopt rsa_keygen_bits:2048

# Generate root CA certificate
openssl req -x509 -new -nodes -key myprivate.
key -sha256 -days 1825 -out root_ca.crt
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank.

For some fields there will be a default value.

If you enter '.', the field will be left blank.

```
--
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-
State]:Colorado
Locality Name (eg, city) []:Denver
Organization Name (eg, company) [Internet
Widgits Pty Ltd]:Rocket Software
Organizational Unit Name (eg, section) []:CSE
Common Name (e.g. server FQDN or YOUR name)
[]:Michael Byrne
Email Address []:support@rs.com
```

You should now see your root CA certificate created in the file root_ca.crt!

Create an SSL Certificate for the Server

Now that we have our root CA certificate, we can move on to creating our first server SSL certificate and sign it with this root CA certificate.

It is important to note, support for the Common Name attribute was deprecated in RFC 2818 since 2000 officially, but for many years browsers did not enforce this setting. However, as of Chrome 58 and Firefox 48, the Common Name value is ignored and you will start getting errors unless you use the subjectAltName x509 extension.

Because we need to add extended X509 attributes to this certificate, we can no longer rely on the prompts and we need to use a config file. As you can see below, I am creating a temporary config file with all the necessary information included that we then supply to the openssl utility. The key part of the extensions is the subjectAltName, this must specify a value with DNS and the name of the server from which HTTPS requests will be served.

```
# Build a config file for the server
certificate
cat &gt; us-l-mb02.conf &lt;&lt;EOF
[req]
distinguished_name = req_distinguished_name
prompt             = no

[req_distinguished_name]
countryName = "US" # C=
stateOrProvinceName = "Colorado" # ST=
localityName = "Denver" # L=
organizationName = "Rocket Software" # O=
organizationalUnitName = "dev" # OU=
commonName = "us-l-mb02" # CN=
emailAddress = "support@rs.com" # CN/
emailAddress=

[v3_ext]
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Custom Generated Server
Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature,
keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:us-l-mb02
EOF

# Create the CSR (Certificate Signing Request)
for server
openssl req -new -key myprivate.key -config
us-l-mb02.conf -out us-l-mb02.csr

# Create SERVER certificate
openssl x509 -req -in us-l-mb02.csr -CA root_
ca.crt -CAkey myprivate.key -CAcreateserial \
-out us-l-mb02.crt -days 1825 -sha256
-extensions v3_ext -extfile us-l-mb02.conf

# Delete the temporary config file
rm us-l-mb02.conf
```

At this point, we have the three pieces we need to now create our server certificate:

- myprivate.key – The private key used to create both the CA Certificate and the CSR

- root_ca.crt – The root certificate we will sign with

- us-l-mb02.crt – The server SSL certificate for our development machine

Using the commands below, we package the CA and server certificate into a PKCS12 certificate. Then we finally create the key store using the keytool utility. The key store (keystore.jks) is what our REST server will use to provide HTTPS for our APIs.

```
# NOTE: You may need to precede openssl commands with winpty on
windows bash("winpty openssl ....")
# Package the CA and server certificates into a PKCS12
certificate
winpty openssl pkcs12 -export -out certificate.pfx -passout
pass:password \
-inkey myprivate.key -in us-l-mb02.crt -certfile ca.crt
# Import the PKCS certificate into the java keystore
keytool -importkeystore -deststorepass password -destkeystore
keystore.jks \
-srckeystore certificate.pfx -srcstorepass password
-srcstoretype PKCS12
```
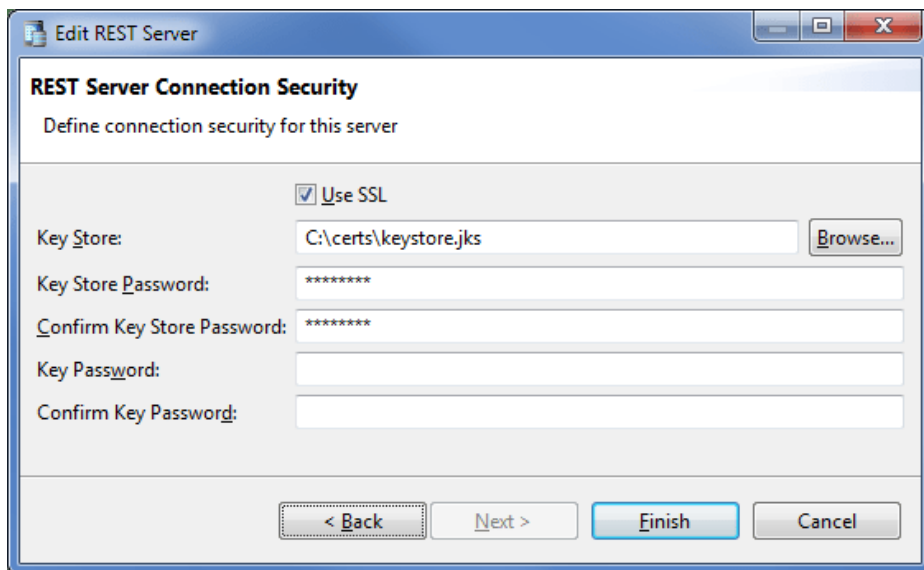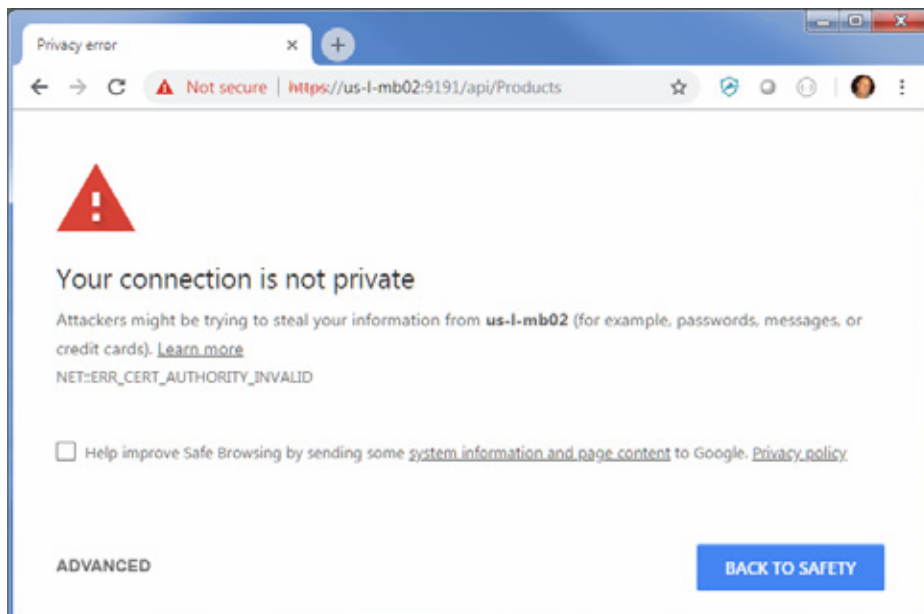

Figure 1

Figure 2

Configure U2 REST Server for HTTPS

At this point, we are ready to configure our U2 REST server to require HTTPS connections. To do this, right-click on your REST server in the RESTful Web Services Developer and choose Properties. Make sure all your information is correct then click Next to get to the REST Server Connection Security panel. Check the Use SSL box and then browse to find where your keystore.jks file is located. If you followed the commands above, enter "password" in the key store password fields. Click Finish to save your changes and then start your REST server by right-clicking and selecting "Start REST Server". Your REST server now requires an HTTPS connection for all connections and uses your key store when clients check which certificates they trust [*Figure 1*].

At this point, if you were to try to hit the REST API with your browser, you will see a page like Figure 2.

This is because we have not configured Chrome to trust certificates that were signed with our root certificate. To do this in Chrome, open a new tab and go to chrome://settings, go to the bottom and click Advanced, then find and click Manage Certificates. This should bring up the Windows Certificate management panel. Click on the Trusted Root Authorities tab and then click on Import. Click Next and then browse to find where you stored your root_ca.crt file. Keep clicking Next and then Finish to import the certificate[*Figure 3*].

Now you're ready to access your REST API securely. Close all your browsers and restart, then go back to your REST URI and you should see the API results as well as the secure lock icon in the upper left of the location bar. Congratulations [*Figure 4*]! **IS**

Figure 3

Figure 4

MICHAEL BYRNE is a Principal Software Engineer in Customer Solutions for the Rocket U2 family of software. Michael specializes in helping customers understand modernization, REST, APIs, mobile, web, and more.
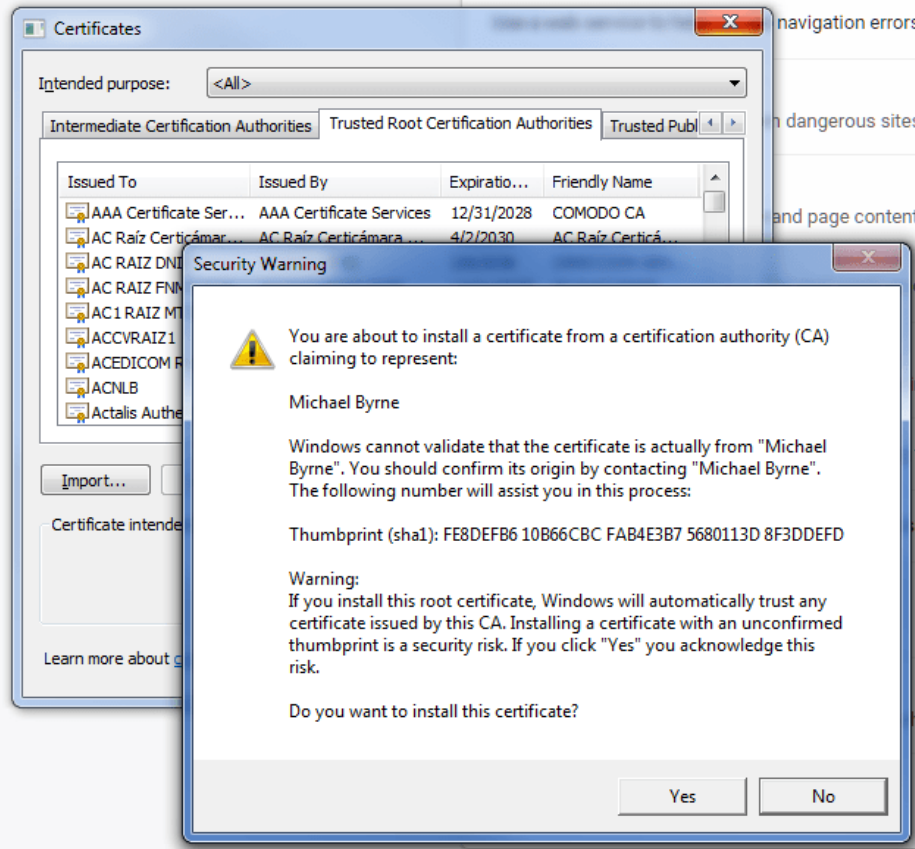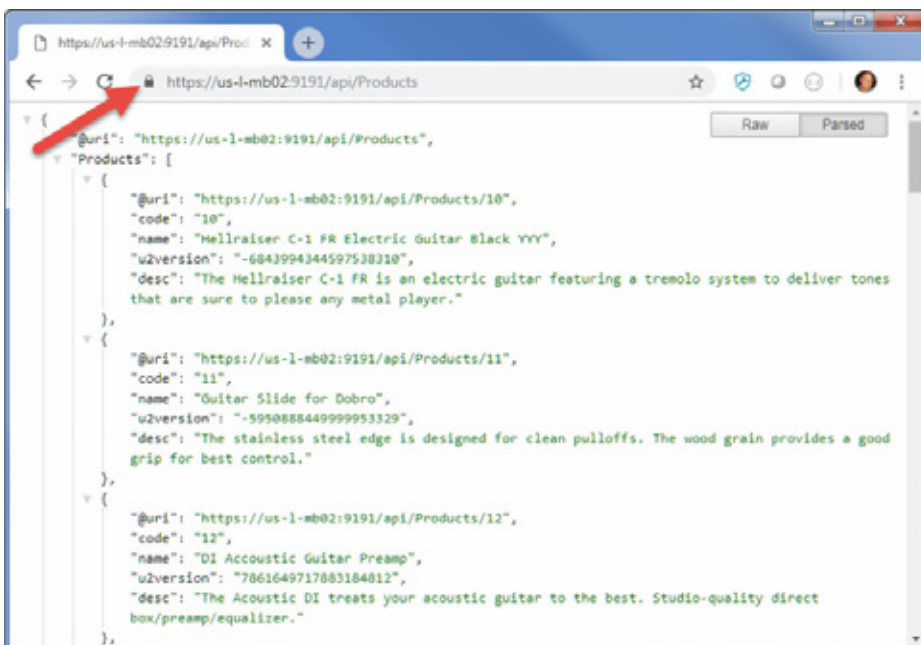
FROM THE PRESS ROOM

Entrinsik Announces Winner of the Informer BI Use Case Award

Entrinsik, Inc. has recently selected the winner of its 2018 Informer Business Intelligence Award to be presented Thursday, September 13th, 2018 at Entrinsicon 2018, Entrinsik's customer conference, in Raleigh, NC. The award is granted to Informer customers for best deployment scenarios that demonstrate the useful and creative ways in which they are using Informer for data analytics, business process improvements, and decision making.

The winner is Terry Thomason from Canadore College. Canadore is a college of applied arts and technology located in Ontario, Canada with over 65 full-time post-secondary programs and over 40,000 alumni. Entrinsik congratulates Mr. Thomason and the team at Canadore who will be giving a special presentation of their use case submission at Entrinsicon. Several other customers will also be presenting their use cases of Entrinsik's Informer and Enrole software.

"The immediate benefits that Canadore has received includes the ability to track applicant conversion rates by school board and individual high-schools which has streamlined our recruitment plan for the upcoming academic cycle. The multiple dashboards that we have developed allowed for much quicker decisions and increased data transparency and collaboration between departments," says Terry Thomason, Institutional Research and Strategic Analyst at Canadore College.

Entrinsik will also be welcoming keynote speaker, Mr. Randy Karsten, the Managing Director of Strategic and Corporate Partnerships at the V Foundation for Cancer Research. The V Foundation is a charitable organization dedicated to saving lives by helping to find a cure for cancer. Founded in North Carolina in 1993, The V Foundation seeks to make a difference by generating broad-based support for cancer research and by creating an urgent awareness among all Americans of the importance of the war against cancer. The V Foundation performs these dual roles through advocacy, education, fundraising and philanthropy. Entrinsik selected the V Foundation because of its North Carolina ties and its tremendous inspiration and hard work to achieve Victory Over Cancer.

Entrinsicon will include breakout sessions with members of the Informer and Enrole product development teams, panel discussions on best practices and methodologies, one-on-one product training, and networking with customers from other organizations. Conference attendees will have the exclusive opportunity to enhance their knowledge of Informer and Enrole software and learn about all the new features, updates, and exciting developments that Entrinsik has planned for the software platforms. ◼

Rocket Invites Young MV Developers Help Guide the Future of MultiValue

Rocket Software hosted a MV GenNext live meetup during the first week of October at the Rocket R&D facility in Denver. The GenNext team includes young MultiValue developers from across the globe who assembled to discuss and collaborate on innovative MultiValue techniques and to talk with product management and engineers about what they wanted to see in the product roadmap.

A significant amount of time was dedicated to roadmap discussion. Each team member voted on categories they wanted to influence. Once the list was narrowed down the team dove in deeper to define the specifics of the requirements they believed were important in each category. The final categories selected were Cloud, Performance, DevOps (defined as a way to automate and monitor all steps of software development), and Development which includes features within the development platform, such as creating an ORM based on standards.

If you are interested in more information on this meet up, read about it here:

https://blog.rocketsoftware. com/multivalue/2018/10/ mv-gennext-young-mv-developers-help-guide-the-future-of-multivalue/ ◼

Kosday Solutions Releases Linkar

Kosday Solutions release LINKAR as a set of professional solutions to help manage license and connectivity with MultiValue DataBases easily and efficiently.

CONNECT MV WITH ALL

• Connect with your DataBase SIMPLY FAST and SAFE.

FROM THE PRESS ROOM

- Share lines. You will OPTIMIZE LICENSES and CONNECTIONS
- Develop ONCE, Do not take care of the DataBase Flavor. LINKAR does it for you.

LINKAR is a Suite of Professional Solutions for Developers that works with PICK MV Databases.

- Linkar Server. Service that is responsive for managing the communication with the MultiValue DataBases.
- Linkar Manager. It's the LINKAR SERVER configuration and monitoring tool.
- Linkar Clients. These are a combination of Libraries that allow the comunication of an application with DataBases. Through its functions the programmer can request and receive data from de DataBase, update it and run any subroutine.

LINKAR WORKS:

- In WINDOWS, LINUX, MAC/OS
- With MICROSOFT .NET, MONO, JAVA, PHP, OFFICE, …

And Connect with OPENQM, D3, UNIVERSE, UNIDATA (coming soon others MV DataBases)

With Functions like RunSubroutine, Execute, Select, Read, Update, New, Conversion, Format….

Use LINKAR libraries in your projects and forget the DataBase flavor.

- LinkarClient.dll
- LinkarClientCOM.dll
- LinkarClient.so
- LinkarClient.jar
- LinkarClientC.dll / LinkarClientC.lib
- libLinkarClientC.so / libLinkarClientC.a
- lksendcommand.exe ∎

Encompass Eliminates Downtime with jBASE from Zumasys

Zumasys and Encompass Supply Chain Solutions, Inc., announced the successful migration of Encompass' custom ERP system to jBASE. The company also migrated its infrastructure from a co-location facility to the Zumasys cloud, now owned and operated by NexusTek.

The move has helped Encompass eliminate costly downtime and streamline development for the company's primary business applications. With its ERP application running on jBASE, Encompass can now take advantage of dramatically improved reliability and performance for batch processing and web apps.

"As Encompass continues to grow our business and service offerings, it is critical to have the right systems in place to run our operations and enable us to scale with confidence," said Encompass President and CEO Robert Coolidge. "We are very pleased with Zumasys' ability to develop a strong, dependable platform to provide exceptional uptime and support for our customers."

Brent Blair, Encompass Vice President of IT, added "We don't have to worry about equipment failures like we had to before because everything is virtual now. "The jBASE application and the interface to our website is much more reliable than our previous solution. We haven't had any downtime as far as our website is concerned since we cut over."

Encompass' business success depends on the company's custom-developed ERP application, which feeds off its PICK MultiValue database platform. As the company's legacy PICK database approached its end of life, it was no longer being actively supported or updated. Over time, the database became slow and unreliable, resulting in downtime for the company's ERP system and eCommerce applications.

Encompass quickly ruled out moving to SAP or Oracle. According to Blair, switching database platforms would have been too costly, risky and time consuming. Worse,

the company would have had to sacrifice the ability to custom tailor its ERP application. "One of the things that makes us competitive in our market is the ability to develop our business application around the specialized needs of our customers," says Blair. "Moving to out-of-the-box software would have been a big loss for us."

The company considered several PICK MultiValue database solutions before deciding on jBASE from Zumasys. With jBASE, Encompass could retain its custom features with a clear path for the future. Zumasys was also the only company that could offer a truly cloud-ready database solution. "By moving to the cloud, we can now focus our resources on development instead of running hardware," adds Blair.

In March 2017, Encompass migrated from its co-location environment to the Zumasys cloud. One year later, the company migrated from its legacy MultiValue system to jBASE in just under four hours. The jBASE migration tool seamlessly migrated 90% of the company's custom code, leaving just 10% of code to transition manually. "It could not have gone more smoothly," adds Blair. "Aside from having to change passwords, most users didn't even realize we had switched systems."

Today, the difference between the systems is clear. With jBASE, processing large

FROM THE PRESS ROOM

transactions takes a fraction of the time that it once did. Processing time for end-of-day and end-of-month transactions has dropped from 3 1/2 hours to less than 1 hour, and performance for the company's website has doubled.

In addition, the system is now more reliable than ever. Adding memory or storage is as simple as an online click or a short 5-minute phone call. Software updates are automatic, so the company never has to worry about downtime for maintenance. Because all data is stored in the Switch Las Vegas, one of the largest and most secure datacenters in the world, compliance and security audits are painless. Customers are thrilled to know their data is protected from security breaches and unwanted intrusion.

"Our customers rely on us to get them the parts they need to be successful," explains Blair. "We are actively investing in solutions to increase reliability and speed for our customers. jBASE gives us a solid foundation for innovation." ∎

Spectrum 2019 Conference Call for Papers

International Spectrum's 2019 MultiValue User and Developer Conference will be at 'The Wigwam' in Phoenix, AZ on April 8-11, 2019.

The aim of the Spectrum Conferences is to bring Users, Administrators, and Developers together to provide solutions, ideas, and enhancements for their MultiValue Enterprise and Business software. The sessions and information provide background and training in how to work with, manipulate, and interact with MultiValue data.

The conference also covers how to enhance and connect desktop, web, mobile, and other user interfaces with existing MultiValue console (telnet/green screen) applications, without loosing the years of investment.

We are looking for recommendations for session topics as well as new speakers. Interested people should submit your request s by Nov 30th, 2019. ∎

Recognizing the Value of MultiValue and its Senior Developers

This is part one of a series on how to bring the MultiValue platform, and the business-critical applications that were built on it, and depend on it, into the mainstream.

The MultiValue community has a wealth of experienced developers responsible for a myriad of applications that have supported businesses for decades, and yet, many of us contribute to the perception that the platform is out-of-date because, as developers, we haven't necessarily kept up with the times.

We have focused so intently on MultiValue that we have isolated ourselves from new technologies and the developers who have adopted them, leading to decreased visibility and, in effect, losing our place at the table. If we:

- Use terms like "old" or "gray-haired" to describe senior MultiValue developers,

- Describe the MultiValue database, applications, and our market as legacy,

- Casually use day-to-day vocabulary that undermines the benefits of MultiValue,

> *You can embrace your expertise and experience while learning how to leverage modern technologies.*

Then, when it comes to promoting MultiValue, we are acting as our own worst enemies.

It's time to reread the IDC Technology Spotlight: In the World of Digital Transformation, Data Is Not Flat: A Fresh Look at MultiValue Database. They were very clear: "The MV DBMS offers significant advantages for a variety of applications." And they concluded that, "Rocket's offerings warrant serious consideration for database applications that require simplicity of development and deployment and ease of operations, but with the data governance features one would expect of an enterprise DBMS." That doesn't sound outdated to me.

So how can you help improve the market's perception? **First, rather than referring to developers who have been in the MultiValue market for a long time as old, let's start using terms like senior developer.** We should recognize senior developers not for their age, but for their experience, as well as their vast knowledge of the business-critical systems that they work on. Often, no one knows more about the nuances of these applications within the organization. They are the development equivalents of Subject Matter Experts (SME). When you consult with an SME, you rarely notice the color of their hair.

Being a Senior Developer in the MultiValue market is not a bad thing. You can embrace your expertise and experience while learning how to leverage modern technologies to enhance your skills and capabilities — and apply them to the tried and true applications that we work on. You can adopt new languages, like Python, so that you can integrate into the more modern workforce.

We need to change our perception of the MultiValue platform and just as importantly, how we view and portray ourselves! MultiValue databases have been around for decades and many applications have not modernized their user interfaces, but that does not mean that we cannot leverage modern development tools and techniques to enhance existing MultiValue applications and create new ones too. It's easy to get started: update the language we use to describe our technology with current terminology. If Senior Developers can't communicate with the new gen-

eration, how will are we supposed to share our wisdom?

As Senior Developers, we need to adjust our thinking, too. If you've never strayed outside your MultiValue comfort zone, you may not realize that modern tools are neither inferior nor a threat. We must learn how to leverage these modern technologies to enhance not only our skills and capabilities, but to apply these technologies to the tried and true applications that we work on.

As a senior developer you should:

- Understand terms and concepts that have been adopted by mainstream databases and related development technologies.

- Know how to talk to younger developers using modern terms rather than the terms that we have used in MultiValue for decades.

- Keep up with newer technologies that can be leveraged with existing MultiValue applications to help mainstream the MultiValue database.

- Abandon the status quo and learn new technologies like JavaScript, Python, PHP and other languages, as well as concepts like leveraging frameworks to reduce the time it takes to develop applications.

It's true, and admittedly discouraging, that young developers are not embracing the MultiValue technology that we love and have committed our careers to: they don't yet know how efficient and effective it is. And it's human nature to be drawn to new, shiny developments. But it's not inevitable that MultiValue and its developers will be passed over by more recently developed database technologies. You just have to embrace your status as a Senior Developer who has much to teach, and much to learn, from developers coming up behind you.

In the next article in this series, I would like to address how we can attract younger developer talent and mainstream the MultiValue technologies that we use. **IS**

DICK THIOT, President/Owner of Maverick Systems, is a technology guru specializing in software development and application design. He supports these skills with a broad knowledge of systems, networks and security. His application development experience centers around database-oriented business applications including accounting systems and information management using MultiValue and SQL Server databases. He has created and continues to specialize in software for automobile dealers and the financial institutions that support them.

M A R K E T P L A C E

ACCOUNTING
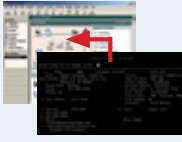
Natec Systems
www.natecsystems.com | nrector@natecsystems.com

COMPLIANCE

SJ+ Systems Associates
www.sjplus.com | sjoslyn@sjplus.com

CONSULTING

Drexel Management Service
www.drexelmgt.com | dconboy@drexelmgt.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.precisonline.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/

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

Philosophy of Indexing

BY CHARLES BAROUCH

I'm in the final stages of writing a new novel, Lefty Lucy. In the story, a medical team is being assembled. One of the prospective members informs the leader that he's a doctor... of philosophy. When asked why he's needed on a medical research mission, he explains the philosophy is needed everywhere. When I wrote that, I started thinking about how philosophy applies to MultiValue. There are a lot of ways I could go with that. As usual for me, I chose to approach it from way out in left field. I started thinking about the real purpose of indexing.

Five Kinds of Data

Before we can talk about indexes, we need an agreed definition of data. For these purposes, I'm going to focus on output formatted data, not the literal representation on disk.

When creating output from any database system, you have four kinds of data available: Raw, Cooked, Computed, and Implied. Each serves a purpose. There is, of course, a fifth kind: Fictional.

> *It is a quasi-magical construct that tracks the information we can extract from the data...*

Raw — This is literally the data as-is. What you recorded to the database is what you retrieve and display.

Cooked — Think of dates and times. The database might store them in some other format — milliseconds or days-since-epoch, for example — so we are transforming them in an extremely unambiguous way.

Computed — This involves performing math on data, concatenating strings, and possibly mixing more than one of the five kinds of data together.

Implied — Let's say you are asked to subset your data and extract just the fifteen records with the highest dollar value. The dollar value is likely computed, but the numbers one through fifteen in column one are implied. The seventh line only has a seven because of

where it lies in the output. You won't find that seven anywhere in the data.

Fictional — While any data can be used in misleading ways, and people lie with statistics all the time, Fictional data is not necessarily untrue. If a client wants me to slap IS for International Spectrum in the third column of every line, so they can identify the source after they merge it with other data, that IS is fictional. It appears nowhere in the actual data, not as-is, not in any derivable format. Data is unlikely to carry a field which indicates that my data is my data.

Quasi-Magical

It is dangerous to generalize about all the flavors of MultiValue, but I believe that every version which supports indexing supports four of the five kinds of data. So, I can index on internal time, on the output of an extended price calculation, or a reassembled full name. Fictional data would also work but, since it is divorced from the effects of other data, all the values in the index would be the same. The odd-man-out is Implied data. I'm not sure how

you'd index on that without an unreasonable amount of brute-force code.

Think about that for a moment. A database index is not merely an index of the content of a file. It is a quasi-magical construct that tracks the information we can extract from the data in addition to indexing the data itself.

Philosophy

When you employ something to make things go faster, you expose a devastating underlying premise: things are too slow when done the right way. In the ideal world, data would always be stored in a form which is identical to the end resulting output. Why don't we do that? Because storing a date as 10-28-2018 makes it slow to sort. Storing the extended price makes it hard to adjust values when an order is amended to or has line-items canceled. We keep the parts and the raw values because of speed and efficiency. Indexes are another symptom of that same viewpoint.

My friend Jim Roberts, who used to work at Applied Digital Data Systems, back when ADDS was an active company in our market, used to tell the story of writing an accounting system in 4k with a soldering iron. When you code that way, you don't have context-sensitive help or error correction. You have code that does the minimal definition of the job. We write all that other stuff because people make mistakes. Imagine how little code you'd need if we did away with prompts, retry loops, and every other bit we wedge in to support being helpful.

The same idea applies to indexes, storing raw data, and every other optimization strategy we employ. We do them

to compensate, to make things that are not inherently fast enough move faster.

So?

If you can find a database structure which doesn't need indexes, I'm confident that you will find some other mechanism protecting some other shortfall. Things are rarely fast enough. Throwing hardware at it helps but even with fast hardware, there is a benefit to code optimization. And, having both of those boosts doesn't mean that better data architecture and better data storage can't be leveraged for even better savings.

When you see clearly which parts of your job are the job -- store the data so it can be retrieved based on criteria, in an order, displaying the correct subset of the data — and the parts which are not the job but support it — indexing and error checking and the like — you can focus your efforts differently.

Cheating?

This is not an article condemning speedups. I heartily endorse them. The hope is that by highlighting the division between our core mission and the things we need to do to support it, we can all get better at explaining the work to others *and* get a new perspective for ourselves.

Next time you're asked why it took so long to do "X" you can split the answer between "X" and the support items. For example: "Building the data tables took a day. Adding help screens with examples, error checking, building in testing tools to prove the data, and creating the reports took three months. I can break that down further for you if you'd like."

Often, the perception that IT is slow comes from people thinking about the core without considering that the ancillary parts take the majority of the time. What we do isn't simple because we aren't just throwing something together. We are building in a way that maximizes the chance that our shiny new thing won't break the first time a breeze hits it. **IS**
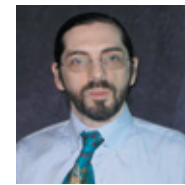
CHARLES BAROUCH is the CTO of HDWP, Inc. and the Publisher at HDWPbooks. You can read his writing in International Spectrum, Theme-Thology, Novo Pulp, Pax Solaria, PerehelionSF, and the Interrogative series, which begins with Tiago and the Masterless.