



Visit us at www.PowerBuilder.SYS-CON.com

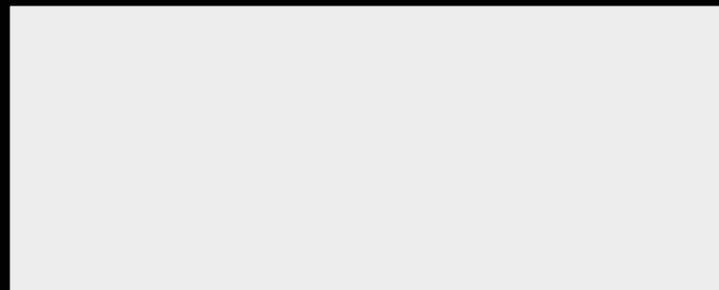
DEVELOPER'S PowerBuilder Journal

JANUARY 2012 / VOLUME: 19 ISSUE 1

Power Building with Exceptions

It's a matter of code elegance

6 MICHAEL ZUSKIN



- 3 Maybe I Haven't Made Myself Clear...
- 12 Perhaps You Can Teach an Old Dog a New Trick Or Two...
- 14 Using the Sybase Hosted Relay Server with SUP 2.1 Hybrid Web Container



PowerBuilder 12.5 Rocking the .NET world... and so much more!

Celebrate 20 years of blazing speed and simplicity

Sybase® PowerBuilder® is, and has been, the award-winning tool of choice for Windows development teams who want results – not excuses. PowerBuilder 12.5 continues this proud tradition with more features and functionality to support all of your development needs. With two Integrated Development Environments (IDEs), PowerBuilder 12.5 is the second full-fledged .NET release, bringing true power and ease of use to .NET.

But it doesn't stop there. It brings fresh new features to your Win32 applications. We also make it easy to migrate your existing Win32 applications to .NET at a fraction of the time, cost and complication of re-writing. So, the choice is yours – innovate in Win32, rock .NET and WPF, or combine the two. Any way you go, you will have the power to create the most updated, visually appealing business applications – on time and under budget.

Join the revolution. Download a 45-day evaluation copy of PowerBuilder 12.5:
www.sybase.com/powerbuilder



Copyright © 2011 Sybase, Inc. All rights reserved. Unpublished rights reserved under U.S. copyright laws. Sybase and the Sybase logo are trademarks of Sybase, Inc. or its subsidiaries. ® indicates registration in the United States of America. SAP and the SAP logo are the trademarks or registered trademarks of SAP AG in Germany and in several other countries. All other trademarks are the property of their respective owners.

PowerBuilder Journal

PRESIDENT & CEO
Fuat Kircaali fuat@sys-con.com

EDITORIAL ADVISORY BOARD
Bruce Armstrong, Loren Corbridge,
Dave Dichmann, Steven Dunn, Sue Dunnell, Berndt Hamboeck, John Olson, Ian Thain

EDITORIAL
Editor-in-Chief
Bruce Armstrong brucea@sys-con.com
Executive Editor
Nancy Valentine nancy@sys-con.com

Research Editor
Bahadır Karuv, PhD bahadir@sys-con.com

ADVERTISING
Senior VP, Sales & Marketing
Carmen Gonzalez carmen@sys-con.com
Advertising Sales Director
Megan Mussa megan@sys-con.com
Sales Manager
Corinna Melcon corinna@sys-con.com

EVENTS
Conference Chair
Jeremy Geelan jeremy@sys-con.com
Conference Program Manager
Krisandra Russo krisandra@sys-con.com
Events Associate
Marilyn Moux marilyn@sys-con.com

PRODUCTION
Art Director / Lead Designer
Tami Lima
Associate Art Directors
Louis F. Cuffari louis@sys-con.com
Abraham Addo abraham@sys-con.com

SYS-CON.COM
Web Designers
Richard Walter richard@sys-con.com

ACCOUNTING
Financial Analyst
Joan LaRose joan@sys-con.com

Accounts Payable
Betty White betty@sys-con.com

SUBSCRIPTIONS
subscribe@sys-con.com
Call 1-888-303-5282
For subscriptions and requests for bulk orders, please send your letters to Subscription Department Cover Price: \$15/issue Domestic: \$149/yr (12 issues) Canada/Mexico: \$169/yr All other countries \$179/yr (U.S. Banks or Money Orders) Back issues: \$12 U.S. (plus S&H)

EDITORIAL OFFICES
SYS-CON MEDIA
577 Chestnut Ridge Rd.,
Woodcliff Lake, NJ 07677
Telephone: 201 802-3000 Fax: 201 782-9601
PowerBuilder Developer's Journal (ISSN #1078-1889)
is published monthly (12 times a year)

Postmaster: send address changes to:
PowerBuilder Developer's Journal
SYS-CON MEDIA
577 Chestnut Ridge Rd.,
Woodcliff Lake, NJ 07677
©Copyright 2012 by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system, without written permission. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.

FOR LIST RENTAL INFORMATION:
Kevin Collopy: 845 731-2684 kevin.collopy@edithroman.com
Frank Cipolla: 845 731-3832 frank.cipolla@epostdirect.com

REPRINTS
For promotional reprints, contact reprint coordinator
Megan Mussa megan@sys-con.com

SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.



Perhaps I Haven't Made Myself Clear...



BY BRUCE ARMSTRONG



I've been discussing HTML5 for some time now. In July of 2010, I mentioned that I wasn't particularly concerned about PowerBuilder supporting HTML5 in the initial PowerBuilder.NET release (12.5) because:

- "HTML5 is largely still in its infancy, and there appears to be too much opportunity for it to fragment as previous HTML standards have done." [1]

In December of 2010 I devoted an entire editorial to HTML5 [2], noting that:

- "I have some basic concerns that make me reluctant to recommend using HTML5 as the basis for any line of business application development in the near future."

But also recommending that as far as a future version of PowerBuilder, Sybase should:

- "Still focus on Silverlight, still work on HTML5 as well, and give us the capability of generating applications that implement both. If the Silverlight player is available, use that, and if not then downgrade to HTML5. That will ensure that we have the capability to deploy apps until such time as HTML5 is mature and capable of being handled by a majority of machines, but also allows the app to work on non-Silverlight enabled devices provided there is a HTML5 browser available to it."

In July of 2011 [3], I again looked at HTML5 and concluded:

- "I expect to see a significant increase in the market penetration of tablet devices in the near future, and that the majority of application development done for those devices will be the development of native apps, not web [e.d. HTML5] apps."

HTML5 also got a passing reference in my editorial for November of 2011 [4], in which I noted:

- "I still believe it's an immature technology."

But also that since Windows 8 is supposed to support the use of HTML5 to generate desktop applications that

- "It's beginning to look like HTML5 may become not only the best long-term bet for web deployment, but for desk-top deployment as well."

Why do I bring all this up? Because in January of 2012 I penned yet another article where I mentioned HTML5 [5], and some people seem to think that last article represented some sort of conversion experience and that I was now a HTML5 fan boy.

So, just to make sure everything is clear, I'd like to review what I said there, compare that to what I said in these earlier articles, and then elaborate a bit further to make sure there's no confusion.

"The primary reason that it even runs well in IE (and only IE 9) is because IE 9 pre-compiles the JavaScript on a background thread"

DOWNLOAD PRINT VERSION

6



- 3 FROM THE EDITOR
Perhaps I Haven't Made Myself Clear...
BRUCE ARMSTRONG
- 6 FEATURE
Power Building with Exceptions
MICHAEL ZUSKIN
- 12 REVIEW
Perhaps You Can Teach an Old Dog a New Trick Or Two...
BRUCE ARMSTRONG
- 14 FEATURE
Using the Sybase Hosted Relay Server with SUP 2.1 Hybrid Web Container
PAUL HORAN
- 22 TECHNIQUES
The PowerBuilder DataWindow as an Image Thumbnail Display Control
BUCK WOOLLEY
- 26 NETWORKS
Big Data in Telecom: The Need for Analytics
BOB HOCKMAN
- 28 MONITORING
Affording the Upgrade: The New World of Network Intelligence
TERENCE MARTIN BRESLIN AND ANDY HUCKRIDGE
- 30 WAN OPTIMIZATION
Virtually Clearing Cloud Initiatives Through WAN Optimization
DR. DAVID HUGHES
- 33 **PBDJ News**

What I said in the January 2012 article, was:

- “While HTML5 may be the future, the future isn't here yet. [...] Nonetheless, the lesson we might take away is that HTML5 **currently isn't suitable** for line of business application development, **though it may be in a few years.**”

I've highlighted a couple of terms in that quote as well as from the previous quotes, because I think they're crucial in understanding what I've been saying for some time. I do not believe that HTML5 is **currently** a mature technology suitable for the generation of enterprise line of business applications. It may be in a few years, it currently isn't. I have to agree with the assessment of Mike James in his iProgrammer article [6] that HTML5 currently is “... another one of those false marketing ideas with very little substance” and that with reference to Microsoft's shift away from Silverlight to HTML5 that it was “...perhaps the most reckless abandonment of a technology in the history of technology.” Don't get the wrong idea from those quotes, Mike goes on to explain why the eventual adoption of HTML5 could be a good thing. He's just noting that somebody needs to “work on the underpinnings that we need to turn it into a usable technology.”

As Mike mentions in his article, and Richard Holdsworth goes into some detail about in his own piece in TechCrunch Europe [7], when you visit an HTML5 website what is really doing the heavy lifting, not HTML5 itself, but JavaScript and CSS [8]. How much of the heavy lifting? Let's take the recently released HTML5 version of Cut the Rope as an example. Of course, Cut the Rope started out as an iPhone application, and it runs quite well on such mobile devices (it was later released for Android and just recently for the Sony Playbook). They recently made it available for HTML5 browsers by porting the OpenGL code over to the HTML5 canvas API and the Objective C code over to JavaScript. How much JavaScript? 15,000 lines running in the browser. [9] And what do I get when I run it in Safari on my iPad? This screenshot shown in Figure 1.

“A little too slow?” You've taken an application that used to run fine on the device, ported it to HTML5, and now it's essentially unusable there. I guess I'm not impressed. I actually get that same message with most other browsers than IE 9 running on a laptop. The primary reason that it even runs well in IE (and only IE 9) is because IE 9 pre-compiles the JavaScript on a background thread.

Don't get me wrong. I do look forward to the day when we can create and run applications that will run on multiple devices in different browsers and will



Figure 1

perform adequately on them all and do that without having to use plugins. HTML5 offers the promise of making that happen. However, it hasn't delivered on that promise yet. When it has, I'll endorse it enthusiastically. Until then, I can only recommend it as something to watch and learn for the future, but not to use for production applications today. ■

References:

1. <http://pbdj.sys-con.com/node/1464997>
2. <http://pbdj.sys-con.com/node/1658103>
3. <http://pbdj.sys-con.com/node/1884245>
4. <http://pbdj.sys-con.com/node/2070202>
5. <http://pbdj.sys-con.com/node/2120922>
6. <http://www.i-programmer.info/professional-programmer/i-programmer/3539-2011-the-year-html5-won.html>
7. <http://eu.techcrunch.com/2011/12/30/html5-idiots-are-confusing-meatballs-with-spaghetti/>
8. When most people refer to HTML5, they are most likely referring to the combination of HTML5, JavaScript, CSS and other web standards (e.g., WebSocket). HTML5 alone would give us very little functionality.
9. <http://www.cuttherope.ie/dev/>

About the Author

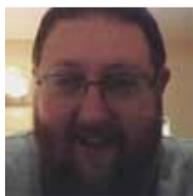
Bruce Armstrong is a development lead with Integrated Data Services (www.get-integrated.com). A charter member of TeamSybase, he has been using PowerBuilder since version 1.0.B. He was a contributing author to SYS-CON's PowerBuilder 4.0 Secrets of the Masters and the editor of SAMs' PowerBuilder 9: Advanced Client/Server Development.

brucea@sys-con.com

Power Building with Exceptions

It's a matter of code elegance

BY MICHAEL ZUSKIN



To pass errors from functions outward, throw exceptions instead of returning an error code like -1. That's implemented by filling the "Throws:" field in the function's header (signature) with the class Exception or its descendant. When the exception mechanism is used, calls to functions look very simple:

```
uf_do_something()
```

As you see, there is no terrible code impurities like

```
if uf_do_something() = -1 then return -1
```

or even

```
li_rc = uf_do_something()
if li_rc = -1 then return -1
```

anymore!

The tradition of returning a success/failure code come from ancient times when exception throwing mechanism

didn't exist yet in PowerBuilder, but there's no need to use horses in the automobile era. We still check code returned by built-in PB functions, as well as in legacy application code, but be a modern person and write new code.

The Art of Exceptions Propagation

The rule of using exceptions in multi-level, multi-branch call hierarchies is simple: if you have a chain of nested function calls (uf_1 calls uf_2; uf_2 calls uf_3; uf_3 calls uf_4; etc.) then the try...catch block usually appears only in the *outermost* function (uf_1) – the calling hierarchies' root, decorating the calling to uf_2; functions uf_2, uf_3 and uf_4 usually don't have their own try...catch blocks – they only transfer exceptions outward. Even more – the root function also can exist without the try...catch block, passing the exception out. A good example of this is a controller NVO in which a root public function, which will be called from another script(s) of your application, calls subsidiary private functions of the NVO. In this situation, the exceptions, having come to the root function from the dark ocean depths of the calls hierarchy, is thrown out by the root function of the NVO instead of returning -1. That will

add elegance to the script that consumes your service, for example, it can be a button's Clicked event – that is really "the first station." So, calling the NVO's public function from it *must* be surrounded with try...catch.

Of course, the not-root functions are not prohibited from having their own try...catch blocks too, but it's a pretty rare practice that happens if they need to process an exception in their special way and then re-throw it.

(To Be) or (Not to Be)?

Calling a function that throws an exception requires special processing in the calling script. Fortunately, the rule is simple: if script A calls script B and script B throws an exception, then script A has two, and only two, choices:

- *To process* (i.e., to catch) the exception. For that, script A must surround the call to script B with a try...catch block.
- *Not to process* the exception (i.e., to pass it outwards) by filling the field "Throws:" in the header of script A). In that case an outer script, calling script A, will bother deciding what to do with the exception.

You don't have to learn this rule: the compiler *forces* you to perform at least one of the described actions, so you have no chance to unintentionally "interrupt current" in nested, multi-level call chains by forgetting to add throwing an exception to the function's signature.

What Is Wrong with the Built-In PB Exceptions?

It's not a big deal to throw an exception in PowerBuilder 8 or later but, IMHO, three very important conditions must be taken into account:

1. The error message, describing the problem, should display the class and the script where the problem occurred.
2. That info must not be typed by the developer each time manually – it should be populated automatically.
3. The code, throwing the exception, must be compact – in fact, it's a piece of technical code, embedded into business code, so it shouldn't be longer than one line. Imagine if your script throws a number of exceptions, and each time an exception object has to be created,

populated and thrown... You'll hardly see the business logic behind all that technical garbage. The following solution is absolutely *not* acceptable:

```
Exception l_ex

try
  [...code...]
  if [condition of Problem 1] then
    l_ex = create Exception
    l_ex.SetMessage("[description of Problem 1] in function
uf_XXX of class n_YYY")
    throw l_ex
  end if

  [...code...]
  if [condition of Problem 2] then
    l_ex = create Exception
    l_ex.SetMessage("[description of Problem 2] in function
uf_XXX of class n_YYY")
    throw l_ex
  end if
catch(Exception e)
  MessageBox("Error", e.GetMessage())
end try
```

How to Solve the Problems

In the suggested solution, three of the actions with the exception (creation + population + throwing) are put in one function named f_throw, so the occupied code shrinks from three lines to one and looks like this:

```
f_throw(PopulateError(0, "[error message]"))
```

As you can see, all the described conditions are satisfied. PopulateError (called inside the argument parentheses of f_throw) grabs the needed details of the thrown exception (class, script and even code line number) and stores them in an Error object. The numeric code, passed as the first argument to PopulateError(), must help the

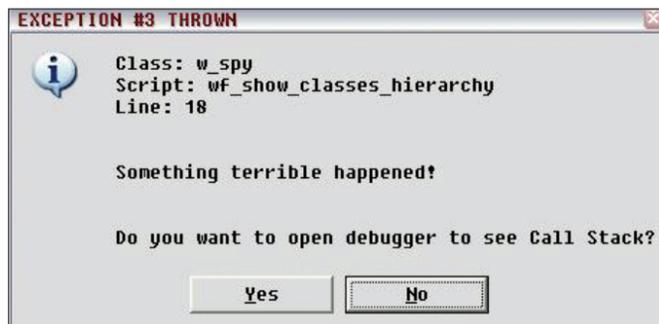


Figure 1

developer find the problem spot when a few exceptions are thrown from the same script; simply pass 0 if you don't need that. The function f_throw creates an instance of exception, populates it with the data stored in Error object, and throws. The function uf_msg, which must be called from within the exception handler, "knows" how to use all that data to build a nice error message. We will use the n_ex, a descendant of Exception, and later I will show how to create that pretty simple class.

```
try
  [...code...]
  if [condition of Problem 1] then f_throw(PopulateError(1, "[de-
scription of Problem 1]"))

  [...code...]
  if [condition of Problem 2] then f_throw(PopulateError(2, "[de-
scription of Problem 2]"))
catch(n_ex e)
  e.uf_msg()
end try
```

Do you see how much shorter that simple example is? And if your script throws a lot of exceptions? And hundreds, if not thousands, all over the application? In the last example the exception is thrown and caught in the same script. It's not a common practice – usually exceptions are thrown and caught in different scripts (i.e., propagate through functions calls). In that case, don't forget to populate the field "Throws:" in the called function's header with "n_ex"!

Example of Use

Let's say, the following code fragment appears in the function wf_show_classes_hierarchy of class w_spy:

```
try
  f_throw(PopulateError(3, "Something terrible happened!"))
catch(n_ex e)
  e.uf_msg()
end try
```

After it has been run, the following message appears (see Figure 1).

As you can see, an additional service is provided: you can open the debugger and "travel" in the functions call stack (this feature is enabled only if the application is running from the PowerBuilder IDE and doesn't work in a standalone executable).

Another Way: Custom Exception Type for Each Problem

The method, described above, utilizes only exceptions of data type n_ex. The method is based on

eLearnIT Step Into The Online Classroom For PowerBuilder

Increase your value with web-based training courses for Sybase professionals everywhere...

Presented by one of the most well known PowerBuilder educators in the world - Mr Yakov Werde - eLearnIT's PowerBuilder Core Skills training course enables software developers at all levels to rapidly develop a comprehensive PowerBuilder skill set.

This course is focused on PowerBuilder deployments in real world scenarios. It will give you the skills to build your own PowerBuilder workgroup scale application from scratch, or perform maintenance tasks on an enterprise scale application. In short, it will increase your value in the workplace, so visit the URL below today!

Each module is presented in an intuitive format that is sure to make the learning experience fun, compelling and rewarding.

Modules covered:

- IDE Skills and Concepts
- PowerScript Fundamentals
- GUI Basics
- Object-Oriented Programming
- DB Connectivity and Embedded SQL
- DataWindow Technology
- Deploying An Application

Working closely with ISUG, eLearnIT has produced a clearly presented product that is tuned to PowerBuilder's most recent releases.

Essential PowerBuilder Series

>> For More Information, visit: www.isug.com/pbtraining



displaying the error message generated in the same code fragment that has thrown the exception. But if you prefer to create exceptions of different types for different problem situations (in the “PL/SQL style”), then our User Manual slightly changes:

1. You have to create the desired set of exceptions (inherited from `n_ex!`), one for each problem to be processed. Of course, their names must reflect their purpose.
2. When you call `f_throw` for those problems, the string, passed as the second parameter to `PopulateError()`, must contain *the name of an exception class to be thrown* instead of *the error message* used in the previous method.
3. A variable, used to catch the exception, must be of the same data type, otherwise the exception will propagate ahead.
4. If the exception's processing includes displaying a message, that message should be generated in the code fragment that processes the caught exception.

Both the methods (“by error message” and “by exception type”) can coexist in the same application. You can create custom exception types for certain problems and still use the first, simple method for the rest of the exceptions.

Here’s an example of using both methods. Let’s say you have created an exception class named `n_ex_credit_prohibited` (don’t forget that it must be inherited from `n_ex`, not from `Exception!`):

```
try
  // The first method:
  if IsNull(as_cust_id) or as_emp_id = "" then f_
throw(PopulateError(1, "No cust_id passed"))
  // The second method:
  if not uf_credit_allowed(as_cust_id) then f_
throw(PopulateError(2, "n_ex_credit_prohibited"))
catch (n_ex_credit_prohibited e1)
  // process the situation when the customer is not eligible to be
credited
catch (n_ex_e_others)
  e_others.uf_msg()
end try
```

Pay attention that the function `uf_msg()` is not used in the second method (the kind of the error is defined by the exception’s type). But if you will call it anyway in that situation, it will display something like “Exception `n_ex_credit_prohibited` thrown.”

How to Create the Related Objects?

You have to perform five simple steps:

1. Create a class inherited from the PB built-in Exception type and save it under the name `n_ex`.
2. Declare in `n_ex` the following instance variables:

```
protected:
  int ii_err_num
  int ii_line
  string is_class
  string is_script
```

3. Add to `n_ex` a public function `uf_populate` with the following code (the arguments are listed in the header comment):

```
/******
*****
Dscr:   Sets data related to the class and the script the excep-
tion is thrown from.
        Called from f_throw().
-----
Arg:    int ai_err_num
        string as_err_msg
        string as_class
        string as_script
        int ai_line
*****
*****/
ii_err_num = ai_err_num
SetMessage(as_err_msg)
is_class = as_class
is_script = as_script
ii_line = ai_line
```

4. Add to `n_ex` a public function `uf_msg`, having no arguments, with the following code (change it if your application uses a specific messaging mechanism, or if you need to register the problem in a log):

```
/******
*****
Dscr:   Displays a message with all the exception-related infor-
mation. Should be
        called for an exception caught in "catch" section of
"try...end try" block.
*****
*****/
int li_user_answer
```

```
string ls_msg
string ls_header

if ii_err_num > 0 then
  ls_header = "EXCEPTION #" + String(ii_err_num) + " THROWN"
else
  ls_header = "EXCEPTION THROWN"
end if

if Len(is_class) > 0 then ls_msg = "Class: " + is_class + "~r~n"
if Len(is_script) > 0 then ls_msg += "Script: " + is_script + "~r~n"
if ii_line > 0 then ls_msg += "Line: " + String(ii_line) + "~r~n"
if ls_msg <> "" then ls_msg += "~r~n~r~n"
```

```
ls_msg += GetMessage()

if Handle(GetApplication()) = 0 /* running from PB IDE, not as .exe
*/ then
  ls_msg += "~r~n~r~n~r~nDo you want to open debugger to see Call
Stack?"
  li_user_answer = MessageBox(ls_header, ls_msg, Information!,
YesNo!, 2)
  if li_user_answer = 1 /* Yes */ then
    DebugBreak()
  end if
else
  MessageBox(ls_header, ls_msg)
end if
```

5. Create a global function `f_throw` (if you don’t like global functions then you can create it as a public function of an object – `uf_throw`) with the following code:

```
/******
*****
Dscr:   Creates an Exception object (of type n_ex or its descen-
dant),
        populates the exception-related data and throws. Is
called this way:
        f_throw(PopulateError(0, "[string]"))

        IF "[string]" is the name of a descendant of n_ex THEN
the thrown exception is of that class;
no error message is generated (the caught
exceptions are distinguished

        by data type, in the "PL/SQL style");
ELSE
        the thrown exception is of the class n_ex;
the string is interpreted as an error message
```

```
which can be displayed

        with uf_msg() of the caught exception;
END IF
-----
Arg:    ai_populate_error_rc_dummy - placeholder for
PopulateError()'s return code;
```

```
the only purpose of this argument is to enable writing

PopulateError inside the argument parentheses of f_throw.
-----
Thr:    n_ex
*****/string
*****/string
string ls_ex_type = "n_ex" //
the default (used if Error.Text contains an error message)
n_ex ln_ex
ClassDefinition lcd
```

```
ls_msg = Trim(Error.Text)
lcd = FindClassDefinition(ls_msg)
if not IsNull(lcd) /* Error.Text contains the name of a valid PB
type */ then
  do until lcd.Name = "powerobject"
    lcd = lcd.Ancestor
    if lcd.Name <> "n_ex" /* the PB type, contained in Er-
ror.Text, is NOT a descendant of n_ex */ then continue
    ls_ex_type = ls_msg
    ls_msg = "Exception " + ls_msg + " thrown." // usually
will not be used, but anyway...
    exit
  loop
end if
```

```
ln_ex = create using ls_ex_type
ln_ex.uf_populate(Error.Number, ls_msg, Error.Object, Error.Obj-
ectEvent, Error.Line)

throw ln_ex
```

That’s it, congratulations – you now have a smart exception class. ■

About the Author

Michael Zuskin is a certified software professional with sophisticated programming skills and experience in Enterprise Software Development.

ursego@gmail.com



Perhaps You Can Teach an Old Dog a New Trick Or Two...

If you've ever had training from Yakov, you'll know he provides a great depth of technical information

BY BRUCE ARMSTRONG



Struggling to get familiar with the new PowerBuilder.NET IDE and the technology it allows you to use (e.g., WPF, REST, WCF and the rest of the alphabet soup)? Then you should run, not walk, over to the free eTutorial series that Sybase just made available for PowerBuilder 12.5: <http://www.sybase.com/powerbuilder125eTutorial>

The training was put together by Yakov Werde. Yakov, for those who may not have heard of him, is the managing partner of eLearnIT LLC (www.elearnitonline.com). Prior to that he spent more than 10 years as a trainer for Sybase, teaching classes on PowerBuilder, Java and .NET. And before that he was a software developer like you and me using PowerBuilder. So he knows of what he speaks. (Yakov is also a regular author for *PowerBuilder Developer's Journal* and the *ISUG Technical Journal*, a member of TeamSybase and a PowerBuilder MVP.)

As the term "eTutorial" implies, this is an online series. All you need to use to take the training is a browser, an Internet connection and monitor, and

some speakers. However, you'll want to have a copy of PowerBuilder.NET (included as part of PowerBuilder 12.5) available at some point to practice what you learn. If you don't already have a copy of PowerBuilder 12.5 you can download a 45-day evaluation version at <http://www.sybasepowerbuilder.com/>



Alternatively, a non-expiring *personal use* version of PowerBuilder 12.5 is one of the benefits of a Green or Gold membership in ISUG, so you may want to consider that as well or instead of the evaluation version: <http://my.isug.com/p/bl/et/blogaid=317&source=1>

Did I mention the training is **FREE**? It is! There is an "in-app" style additional purchase option for the lab exercises at a very reasonable price. I'd highly recommend that. There's only so much you can learn from listening to someone lecture, no matter how good they are. You learn significantly more if you go hands on and put into practice what you're learning. I think the lab exercises are a must if you really want to follow Yakov through the practical application of the material he's covering.

I must also warn you though, the material is extensive. While that's a good thing, there is one downside. In some of the previous training courses that Yakov has put together, one very nice feature is that you could stop the training at some point and then pick up where you left off some time later, even on a different machine. Given that you don't log in to this training, I suppose it would be difficult for it to remember who you are to support that. There is a bookmarking feature, so that you could capture a link to the location you're stopping at. But that's a much more manual process (you have to remember to do it), and I found that while it works in Internet Explorer, I couldn't get it to work with Firefox or Chrome.

Aside from that minor annoyance, the material is great. Yakov was always one of the instructors that you hoped would be doing your sessions when you were taking classroom sessions, and I'm very glad he's still actively involved in providing PowerBuilder training.

Did I mention it was extensive? My summary of the duration (in minutes) of the various sections as follows:

- o IDE and Methodology 115
- o Cloud Services and REST 60
- o WPF
 - Overview 15
 - XAML Syntax 30
 - Layout Managers 54
 - Controls 20
 - Data Binding 30
 - Resources 30

That's roughly 6 hours of training, minus the lab exercises. That might sound like something you could sit through in one day, but I don't think you could really do it without your brain exploding. I

found I was only able to give it about a half hour to an hour a day just because of my hectic schedule. Of course the great thing about an eTutorial is that you can do that. Just give it whatever time you have and as often as you can. The instructor is always there ready to pick up where you left off. If this was a classroom-led instruction, I'd estimate that there's at least a week's worth of material here (assuming a couple of hours training in the morning and lab exercises in the afternoon).

If you've ever had training from Yakov, you'll know he provides a great depth of technical information, not just regurgitating the user manual at you. I think I was particularly impressed that he included a demonstration of how to use the Fiddler proxy tool (a free utility from Microsoft) to debug web services. That's not something a lot of instructors would think to include. In a number of places there are references that you can follow to gain additional information. Sometimes it's a book recommendation and clicking on the picture of the book will take you directly to Amazon.com if you wish to purchase it.

What's this all have to do with old dogs? Well, to tell you the truth when I went to take a look at the eTutorial I was expecting to see how good it might be for training others. I wasn't really expecting to learn anything from it myself. I've been working with .NET since 2003 and with Visual Studio since before there was a .NET (primarily C++). One of my main focuses for the last several years has been web services, so I'm quite comfortable with WCF and REST. Although it hasn't been one of my main focuses lately, I've had enough experience with WPF that I wasn't expecting to get much out of an introductory lesson. Well this old dog learned quite a few new tricks from this series. If I can learn something from it with as much experience as I have with these technologies, you certainly stand to learn a lot from this training if you are new to all of it.

It's free (aside from a nominal charge for the lab exercises) and it is extensive. What are you waiting for? ■

About the Author

Bruce Armstrong is a development lead with Integrated Data Services (www.get-integrated.com). A charter member of TeamSybase, he has been using PowerBuilder since version 1.0.B. He was a contributing author to SYS-CON's *PowerBuilder 4.0 Secrets of the Masters* and the editor of *SAM's PowerBuilder 9: Advanced Client/Server Development*.

brucea@sys-con.com

Using the Sybase Hosted Relay Server with SUP 2.1 Hybrid Web Container

The iAnywhere Relay Server

BY PAUL HORAN



This article will cover the steps necessary to use the Sybase Hosted Relay Server with your local install of SUP 2.1.

The iAnywhere Relay Server

The iAnywhere Relay Server is an optional (but critical, IMO) component of the Sybase/SAP mobile infrastructure. It's essentially a reverse-proxy server that gets installed into a corporate DMZ, and implements an "inbound port" model to help insulate the server(s) inside the firewall from malicious internet traffic. It can be used with any of our mobile solutions, including SUP, Afaria, SQL Anywhere, and MobiLink. Mobile devices don't need to be configured with the addresses/IPs/ports of the SUP server or farm – they only need the address and port of the Relay Server. The SUP server(s) makes an outbound connection to the Relay Server using regular HTTP/S traffic, meaning there are no open ports through the inner corporate firewall. Devices make an inbound connection to the Relay Server (again, HTTP/S on ports 80 or 443), and the Relay Server brokers the conversation between device and server, much like an old-time telephone operator at a switchboard. If the Relay Server is not implemented, the only other option to get devices and servers to communicate is to open specific TCP/IP ports for the SUP server through the inner and outer firewalls, and to configure the devices with the addresses of those SUP servers. I've not yet met a network admin that would readily agree to this...

The Relay Server is included free-of-charge with SUP, Afaria, and SQL Anywhere, and runs inside Microsoft IIS or Apache web servers.

The Hosted Relay Server

Sybase has created a hosted implementation of the Relay Server, which is available for free to registered Sybase/SAP customers. This is designed to allow for development and test of small mobile deployments – it's not to be used for production implementations (for obvious reasons). The URL for the hosted relay server is: <http://relayserver.sybase.com>. The first step is

to submit a registration request, which will be reviewed by Sybase. Once approved, there are three main steps to connect a remote device to the SUP server farm.

1. Register the SUP Farm details

At the bottom of the account details page, you'll find the individual hyperlinks to create the different types of server farm registrations. We'll be creating an SUP farm. The Relay Server communicates with an SUP server "farm," even if there's only a single server node registered in that farm. This makes it easy to add and remove nodes from the farm, without having to reconfigure any of the mobile devices out in the field.

Click "Add New Sybase Unwired Platform Farm", and this brings up the page shown in Figure 1.

I've placed the Relay Server registration page over the Sybase Control Center page, so it's easier to see how to fill out the form. Since this example will focus on the Hybrid Web Container, we'll create a Message-based Synchronization (MBS) farm. I'm also registering a Replication-based Sync (RBS) farm, even though this example is going to focus solely on the MBS capabilities of the Hybrid Web Container.



Figure 1

“The iAnywhere Relay Server is an optional (but critical, IMO) component of the Sybase/SAP mobile infrastructure”

“Sybase has created a hosted implementation of the Relay Server, which is available for free to registered Sybase/SAP customers”

Give the farm a unique name. As you can see, I've called mine **PHoranXP**. The Relay Server will automatically append a suffix to the farm id, based on its type. The full name of the farm is composed as Registration Name (dot) Farm ID + suffix. Message-based farms get the letters MBS added, so its official name will be **PaulHoran.PHoranXPMBS**. The Replication-based Sync farm will be named **PaulHoran.PHoranXPRBS**. If I'd created a DCN farm, it would have been **PaulHoran.PHoranXPDCN**. Bottom line: don't forget the registration name or the suffix...

Last, enter the server node name within the farm. This is comprised of the SUP server name that appears in the Sybase Control Center, plus the string "SUPServer1." Mine happens to be named **phoran-d630SUPServer1**. Mine happens to be named **phoran-d630SUPServer1**. Click Create Farm to register the details to the Relay Server, and create a long alphanumeric "token" value for each farm type. You'll be asked for this value in step 2 below. If you ever forget this value (and who could be expected to remember it?), just click on the Configuration Instructions link, and you'll be taken to a page that lists all the pertinent info for your registration.

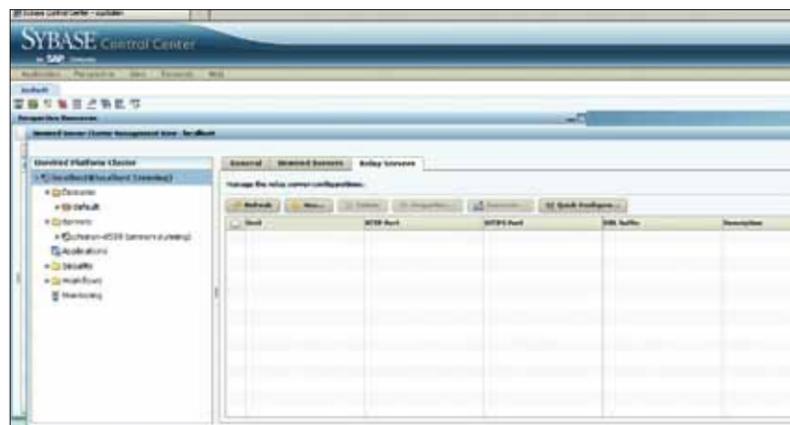


Figure 2

The remaining steps are performed in the Sybase Control Center (SCC), so open a session to the SUP server now.

2. Point SUP to the Hosted Relay Server

The SUP server establishes an outbound connection to the Relay Server through a component called the Relay Server Outbound Enabler, or RSOE. SUP needs to know the connection details to the Relay Server, and they're configured in the Sybase Control Center. This is configured in the main server properties panel of SCC, in the Relay Servers tabpage. Figure 2 shows what this looks like.

Click the **New...** button to open the first screen of the registration wizard. Fill in the following values:

- **Host:** relaysvr.sybase.com
- **HTTP port:** 80
- **HTTPS port:** 443
- **URL suffix:** accept the default value
- **Description:** your choice...

Click the Next button to navigate to the next panel in the wizard. This screen is a little more complicated, and is a two-step process. Figure 3 shows both of the wizard screens, already filled out.

We're now working in the top half of the second screen. Enter the full FarmID that was registered to the specific Relay Server/Type combination in step 1. Since I'm registering a messaging server, I'll select **Messaging** in the Type field, and enter **PaulHoran.PHoranXPMBS** as the farm ID. Enter a description, if desired. Now click the yellow plus sign to register that combination. To create a second registration (for a Replication or DCN type), you would simply change the Type field to the desired value, change the suffix in the farm ID, and click the yellow plus sign again. If you make a mistake, simply select the row, and click the red X to delete it and start over.

Now we move to the bottom half of the screen, where you're registering the server node within the farm ID and type. Select the appropriate row in the top list (I only have one, but you might have an MBS and an RBS row registered). Since this is a Master-Detail screen, it's important to keep the relationships straight. Enter the Server Node ID and the corresponding Token value for that node in the corresponding fields, then click the **BOTTOM** yellow plus sign. (If it's grayed out or disabled, make sure you have the correct row selected in the top half of the screen.) I've entered **phoran-d630SUPServer1**, and pasted in the correct token value from the Hosted Relay Server configuration page.

When everything is done correctly, your screen should look somewhat like the right side of Figure 3 above. Click Finish to record the registration and close the wizard.

3. Create the RSOE details

Figure 4 below shows how to find the panel for recording the properties of the Relay Server Outbound Enabler (RSOE) process.

Begin by clicking the yellow plus sign. This opens a three-panel wizard dialog for entering the properties, as shown in Figure 5. The good news is this one is much easier to navigate. Select Messaging in the Farm Type field, and all the other properties will be automatically filled in.

The only modifiable options on the first screen are the port values. The RSOE process connects to the SUP server on port 5001, and out to the Relay server on port 80 or 443. Clicking Next> allows you to provide HTTP credentials (if necessary), and specify an HTTP proxy. The third and final screen allows for setting specific processing and logging options. These can all be modified later if necessary, so just accept the defaults and click Finish to complete the process.

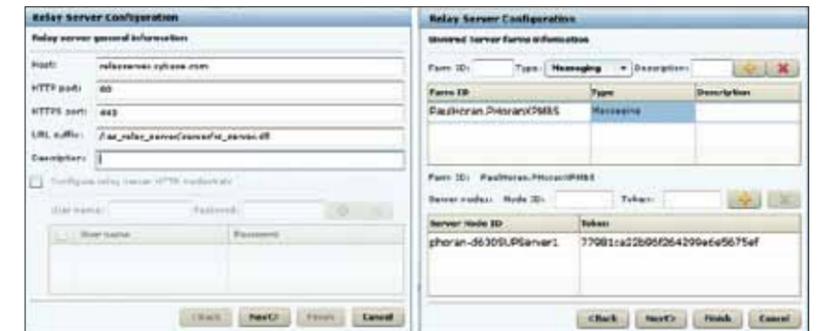


Figure 3

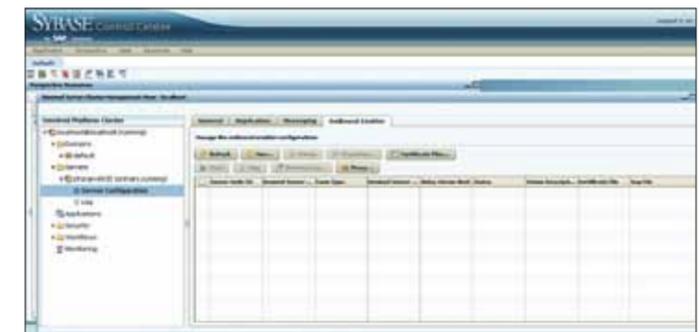


Figure 4

The last step is to actually start the RSOE process. If registration was successful, a row should appear in the RSOE list. Select it, and click the Start button. Wait several seconds, then click the Refresh button to see the status of the RSOE process. The Status field should show as "Running".

Configuring the SUP Container on Mobile Devices

Registering Users in Sybase Control Center

The first step in hooking up a device to SUP through the Hosted Relay Server is to register that

“The SUP server establishes an outbound connection to the Relay Server through a component called the Relay Server Outbound Enabler, or RSOE”



Figure 5



Figure 6

username in SCC. This is done in the Application Connections tab of the Applications panel. Figure 6 shows the registration for my iPad device.

Note that the server name is set to **relayserver.sybase.com** on port 80 – not the default of **phoran-d630** on port 5001. Set the Application ID to HWC to 1 link the user to the Hybrid Web apps. Give the user an appropriate activation code and click OK.

Configuring the Connection Properties on the Device

The SUP Hybrid Web Container for iOS 4.x/5.x can be downloaded directly from the Apple Appstore (<http://itunes.apple.com/us/app/syb->

“The first step in hooking up a device to SUP through the Hosted Relay Server is to register that username in SCC. This is done in the Application Connections tab of the Applications panel”

ase-mobile-workflow-2.1/id474380077?mt=8). The first thing you’ll do after downloading and installing it is configure the connection properties in the Settings app.

- **Server Name:** relayserver.sybase.com
- **Server Port:** 80 (or 443, if you configured the secure HTTP connection)
- **Farm ID:** the entire farm ID, including the registration name and suffix – mine is PaulHoran.PHoranXPMBS
- **User Name/Activation Code:** the username and activation code that were registered above.
- **Password:** provide a 6-digit password, or leave this blank to force the user to reenter the HWC password each time the app is reactivated.

Once these connection properties are set and the Hybrid Web Container is launched, a connection is made to SUP through the Relay Server to download all the registered Hybrid Web apps that have been assigned to the device user.

Resource

The Relay Server: http://www.sybase.com/files/White_Papers/The-Relay-Server-WP.pdf

...

This article was republished from Paul Horan’s blog: <http://blogs.sybase.com/phoran/?p=491>.

About the Author

Paul Horan is a Principal Systems Consultant at Sybase. He has a wide range of experience with Sybase products, including PowerBuilder, PowerDesigner, Sybase Unwired Platform, SQLAnywhere / MobiLink, Afaria, and Mobile Office. He works out of Sybase’s Bethesda MD office, and was inducted into the ranks of TeamSybase in 2000. A 1984 graduate of Indiana University, Paul currently resides in Arlington VA.

DEVELOPER'S PowerBuilder Journal

Is Just a Click Away...



Subscribe Today!

www.PowerBuilder.SYS-CON.com



PowerBuilder Developer's Journal (PBDJ) is the only monthly source of quality PowerBuilder information since 1994.

SPECIAL SHOW OFFER! SAVE 35%*

1-Year Print & Digital Edition Subscription for 12 issues
ONLY \$64.35 (Reg. \$99)

2-Year Print & Digital Edition Subscription for 24 issues
ONLY \$83.85 (Reg. \$129)

1-Year & Digital Edition PDF Subscription for 12 issues
ONLY \$44.85 (Reg. \$69)

FREE
6 Month
Digital
Edition
Trial

OFFER SUBJECT TO CHANGE WITHOUT NOTICE



Celebrate 20 years of blazing speed and simplicity

Twenty years ago, PowerBuilder revolutionized the application development world with its speed, simplicity and power. Anyone who has used its legendary DataWindow will tell you it is still the fastest and easiest component for data access around. And over the years, PowerBuilder has evolved to stay in synch with the latest functionality and visual user interface advances.

PowerBuilder 12.5 continues this proud tradition with more features and functionality to support all of your development needs – with two Integrated Development Environments (IDEs) that give you options. PowerBuilder 12.5 is the second full-fledged .NET release, bringing true power and ease of use to .NET, including a new feature that allows you to build Custom Visual User Objects that encapsulate the DataWindow and share them with your .NET developer peers.

But it doesn't stop there. PowerBuilder 12.5 brings fresh new features for your Win32 applications as well. And we also include an incredibly fast and cost-effective migration path to evolve your Win32 applications to .NET. There's no need to toss out your PowerBuilder code to get your application to .NET. PowerBuilder gets you there, while protecting your investment. So, the choice is yours – innovate in Win32, Rock .NET and WPF, or combine the two. Any way you go, you will have the power to create the most updated, visually appealing business applications – on time and under budget.

Download a 45-day evaluation copy of PowerBuilder 12.5 at www.sybase.com/powerbuilder

SYBASE
An SAP Company





The PowerBuilder DataWindow as an Image Thumbnail Display Control

Using a metadata technique of storing multiple properties in a single string

BY BUCK WOOLLEY



While external controls exist for viewing image thumbnails, there can be the need to view images as thumbnails within the context of other data within a multi-row list, treeview or report. This can be easily done with a PowerBuilder DataWindow object. In this case a treeview DataWindow (see Figure 1) was needed to display multiple images as thumbnails that related to the data presented in each row. Required functionalities include making the images selectable, displaying data for each thumbnail, such as the name of the file, and to keep the appearance orderly by having the images appear the same height in a row.

I will present the techniques used to create thumbnail images in DataWindows. The methods presented assume that the images are stored as files in some folder that's accessible by the application. It's also assumed that the image types are any of the standard

image types that are recognized by PowerBuilder: .bmp, .gif, .jpg, .wmf, and .png. The image files are not converted in any way but are displayed in a thumbnail size.

Overview

The techniques for displaying thumbnails in a DataWindow are based on my previous work on DataWindows ("Utilizing the Power of the DataWindow Object," *ISUG Technical Journal* Q1 2006). This technique basically uses a metadata technique of storing multiple properties in a single string, in this case storing the image display properties in different fixed length sections of a string. These property strings are stored in DataWindow columns, so that each row can contain its own set of images along with all the properties to properly display each image. The columns are used to store the values for each image property such

“While external controls exist for viewing image thumbnails, there can be the need to view images as thumbnails within the context of other data within a multi-row list, treeview or report”

as *x* and *y* location, width, height, visibility and image file name. These properties are used in a DataWindow by addressing the corresponding section of the string for each graphic object and converting it into a string using the MID(string,start,length) function or number if necessary using the LONG(MID(string,start,length)) function.

Getting Started

The base DataWindow should contain string columns to hold the properties for each image. The strings should be long enough to contain all the data required for the maximum number of images for each DataWindow row. In this case 40 is the maximum number of images that would be displayed per row so each column had to be defined as 40 times the length of the property for each image. Numeric properties such as *x*, *y*, width can each have a maximum length of 5 so they must be defined in strings that are 200 characters long. Visibility only requires 1 character, either '0' or '1,' so it can be defined in a string of length 40. The fully qualified names of image files can vary greatly so the maximum length should be considered when creating the DataWindow. In this case the maximum length is 100 characters, which requires a column that is 4000 characters wide.

Figure 2 shows the contents of the different property columns. The *x*, width, visibility and file name properties for the first three images are highlighted with different colors corresponding to the border color of the actual images. You can see the relationship between the image and each of its properties.

These columns are defined in the DataWindow during development, either in an external or select DataWindow. In this case they are defined in the select statement for the DataWindow:

```
SELECT *** , '' as x, '' as w, '' as filepath, '' as vis, '' as y,
'' as filename, ***
FROM ***
WHERE ***
```

The lengths of the fields are modified during runtime. The vis column, which contains the parameter for image visibility, is set to FILL('0',40) so that initially no images are visible for any row.

At Runtime

The actual image objects are defined dynamically once at runtime during the startup process. Since it's unknown how many images are needed, the maximum number is created even if it's highly unlikely that they will all be used. A text field is also created to display the name of the file. A loop is used to create 40 instances of the compute object that will display the image:

```
dw_cmd = dw_cmd + ' create compute(band=detail alignment="2" ex
pression="bitmap(trim(mid(filepath,'+STRING(l1_x100)+' ,100)))"
border="1" color="0" x="0-tlong(mid(x,'+STRING(l1_x6)+' ,6))"
y="0-tlong(mid(y,'+STRING(l1_x6)+' ,6))" height="200" width="0-tlon
g(mid(w,'+STRING(l1_x6)+' ,6))" format="[GENERAL]" html.valueisht-
ml="0" name=p'+STRING(l1_row)+' visible="0-tlong(mid(vis,'+STRING(
l1_x1)+' ,1))" *** )'
```

```
y="0-tlong(mid(y,'+STRING(l1_x6)+' ,6))" height="200" width="0-tlon
g(mid(w,'+STRING(l1_x6)+' ,6))" format="[GENERAL]" html.valueisht-
ml="0" name=p'+STRING(l1_row)+' visible="0-tlong(mid(vis,'+STRING(
l1_x1)+' ,1))" *** )'
```

A compute object is used to take advantage of the bitmap() function that converts a string into an image. The name of the object contains the sequence of the properties for that object, for instance p1, p2, p3. The sequence number will be stripped from the name and used to point to the location in the string that contains the properties for that particular object. Please note the items in bold. These are the properties that are held in the string columns that were defined in the DataWindow. There are four variables that control where in the string the property should appear. They are incremented by the same amount for each pass in the loop:

```
l1_x1 = l1_x1 + 1
l1_x6 = l1_x6 + 6
l1_x50 = l1_x50 + 50
l1_x100 = l1_x100 + 100
```

The result is 40 identical compute objects that refer to different sections of the property columns. This section



Figure 1: Thumbnail images within a treeview DataWindow

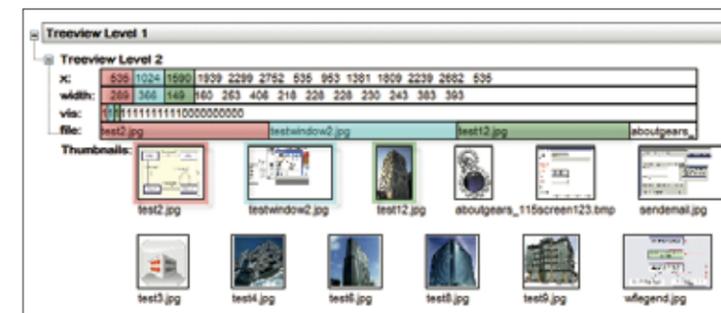


Figure 2: Screenshot showing the formatting of various display property columns in the DataWindow. The background color of the data 'cells' correspond to the colors of the frames surrounding the thumbnail image.

“Using some advanced functionality the DataWindow control can be used as an image thumbnail display tool. The techniques described can also be embedded in reports to display images in a report”

of code shows the definition of the second image, p2, that was created at runtime from the code above. The variables have been replaced with values of the location within the string for each property:

```
' create compute(band=detail alignment="2" expression="bit
map(trim(mid(filepath,101,100)))" border="1" color="0" x="0
long(mid(x,7,6))" y="0 long(mid(y,7,6))" height="200" width="0
long(mid(w,7,6))" format="[GENERAL]" html.valueishtml="0" name=p2
visible="0 long(mid(vis,2,1))" *** )'
```

For instance for image p2 the property expression="bit map(trim(mid(filepath,101,100)))" indicates that the function bitmap() should look at the text that exists starting at position 101 for a length of 100 in the column filepath. This will return the fully qualified name for the image file. The same process exists for the other display properties.

When the DataWindow is displayed another datastore retrieves all the images for the DataWindow. A loop is set up to build the property strings for each row in the presentation DataWindow. A filter is used to isolate the image properties for each row in the presentation DataWindow. If images exist for a row, a loop is run to extract the prop-

erties for each image that are concatenated together into variables on a fixed length basis.

```
ls_image_file = lds.object.filepath[l1_row]
ls_image_nsme = lds.object.filesme[l1_row]
//set the qualified file name to a string of a length 100
ls_file = ls_file + LEFT(ls_image_file +FILL(' ',100),100)
//set the file name to a string of a length 50 to display under the
image
ls_filename = ls_filename + LEFT(ls_image_name[l1_row]+FILL('
',50),50)
//make the image visible
ls_vis = ls_vis+'1'
```

Once complete, these variables are then loaded into the appropriate columns in the display DataWindow. As you can see in the figures the width of each image varies but the height of each image is constant and is set to 200 units. To keep the proportions of the original image the width must be converted. This is done by moving the image file to an invisible picture object located on the same window as the presentation DataWindow, called p_1, that is also 200 units in height.

```
//set the name from the DataWindow
ls_image_file = lds.object.filepath[l1_row]
//set the picturename property on the picture object
p_1.picturename = ls_image_file
//set the originalsize = TRUE to get the correct dimensions
p_1.originalsize = TRUE
//set height to 200
l1_height = 200
//get the ratio of desired height to the picture height
ld_ratio = l1_height / p_1.height
//set the thumbnail width to the image width * the ratio of the
heights
l1_width = p_1.width * ld_ratio
//concatenate the calculated width to the width property string
ls_width = ls_width + RIGHT(' '+STRING(l1_width),6)
//concatenate the x value to the x property string
ls_x = ls_x + RIGHT(' '+STRING(l1_x),6)
//concatenate the y value to the y property string
```

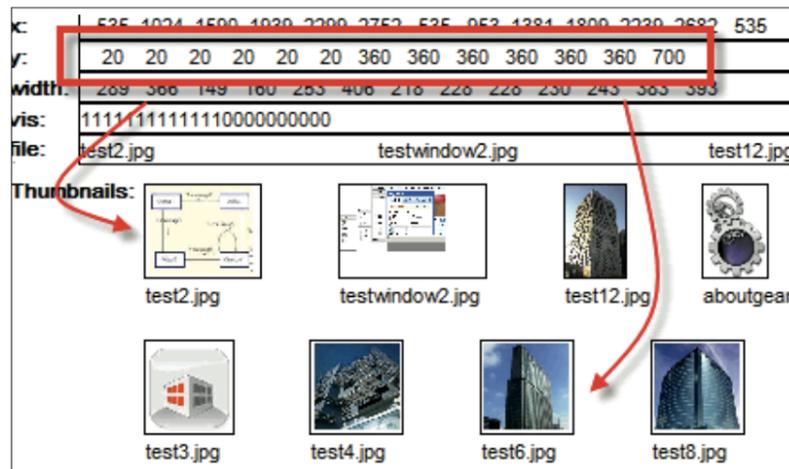


Figure 3: Screenshot showing the values that hold the y location property for the thumbnail images.

```
ls_y = ls_y + RIGHT(' '+STRING(l1_y),6)
//calculate the x value for the next image by adding the width of
the current image plus 200 for extra space between images
l1_x = l1_x + l1_w + 200
```

If the images need to appear larger, simply change the height constant to a larger number. This can actually be set at runtime before the DataWindow is displayed.

Another issue that needs to be handled is setting the x location of each image. Theoretically the x value can be incremented up to over 32,000, but this would generate a very wide DataWindow that would require a lot of horizontal scrolling. To make things more user-friendly it would be better to create the images in a format that appears as rows of images. These rows are not actual data rows in a DataWindow but the images are displayed to appear in a row format within the DataWindow detail or trailer band (see Figure 3). This is achieved by changing the y value after the x value is greater than a certain amount. In this case the maximum x value is set to 3000. This can be adjusted at runtime based on the width of the DataWindow. When the x value is greater than the max value, the x value is reset to the starting value of 535 and the y value is increased by 340.

```
l1_x = l1_x + 200 + l1_w
IF l1_x > 3000 THEN
    l1_y = l1_y + 340
    l1_x = 535
END IF
```

Selectability

The images are selectable with the mouse by using the doubleclicked() event. A user may want to select an image to open a detail window or to view a larger version of the image. In this case the image is selected and passed as a parameter to an external imageviewer control (see Figure 4). In the DataWindow's doubleclicked() event the getobjectatpointer() function is used to get the object under the cursor. The object name contains the sequence of the image in the row. For instance object p2 is image number 2. The sequence is extracted using various PowerBuilder string functions. The location of the file name in the filepath column is extracted using the sequence number.

```
//get the object
ls_object = this.getobjectatpointer()
//extract the row of the object
row = long(right(ls_object,len(ls_object) - pos(ls_object,'~t'))
//extract the name of the object
ls_object_name = LEFT(ls_object,POS(ls_object,'~t') - 1)
//extract the sequence of the object
IF ISNUMBER(MID(ls_object_name,LEN(ls_object_name) - 1,1)) THEN
    l1_seq = LONG(RIGHT(ls_object_name,LEN(ls_object_name) - 2))
ELSE
```



Figure 4: Demonstrating selectability by double clicking on a thumbnail to open the full size image in an image viewer.

```
l1_seq = LONG(RIGHT(ls_object_name,LEN(ls_object_name) - 1))
END IF
//get the string containing the file names of all images on the row.
ls_filepath = this.object.filepath[row]
//extract the file name of the object that was selected
ls_file = MID(ls_filepath,(l1_seq * 100) - 99,100)
//call the imageview with the file name
command='rundll32.exe shimgvw.dll,ImageView_Fullscreen '+ ls_file
Run(command)
```

The same function can also be called from the clicked() event and the rbuttondown() event to control a dropdownmenu display.

Conclusion

Using some advanced functionality the DataWindow control can be used as an image thumbnail display tool. The techniques described can also be embedded in reports to display images in a report. The examples and code presented in this article were created using PB 12.5 classic; however, the same techniques have worked since at least PB version 6.5. Applications using the same techniques have also been successfully deployed to Apeon as well as created in Visual Basic using DataWindow.NET. It's assumed that these techniques can be deployed in PB.NET as well.

About the Author

Buck Woolley has been involved in programming and software development since 1987 and has been a Powerbuilder consultant since 1994, starting with version 3.0. Currently he is the main client server developer at Patrix AB in Gothenberg, Sweden. He has made numerous Techwave presentations covering advanced datawindow techniques. Buck and his wife reside in San Diego, California.

bwoolley@dw-extreme.com



Big Data in Telecom: The Need for Analytics

Translating raw data into actionable intelligence delivers value for the enterprise and the customer

BY BOB HOCKMAN



Networks have become a strategic asset, the life blood of organizations. Once considered a “techy thing,” networks are now mission-critical for every member of the organization – from the IT manager to the marketing VP to the CEO. An increasing number of companies now recognize the impact network quality has on the customer experience and, in turn, on the bottom line.

Providing a great customer experience, every time, is vital for limiting churn and building loyalty. This has led many organizations to adopt a strong quality assurance program to test and monitor all contact center services. This is particularly important in environments that must support multi-channel and multi-service applications. The complex configurations needed to enable voice, video and data to share network resources puts a tremendous strain on bandwidth and creates problems that can be very difficult to isolate. And the proliferation of mobile devices has added an entirely new set of issues and customer behaviors.

Network testing and quality assurance monitoring is important for creating and maintaining a great customer experience. However, these solutions capture a tremendous amount of information on customers, networks and operations. Managers constantly create reports with a slew of KPIs, including average hold time, peak chat volumes, queue lengths and IVR satisfaction rates, to try to make sense of all the data. While this provides a certain amount of understanding of daily operations, companies can do better. Advanced network analytics that correlate and trend

multiple data points and KPIs over multiple dimensions offer real insight that companies can use to gain greater control over expenses, resolve issues more quickly and plan more effectively.

Using Analytics to Troubleshoot

Troubleshooting issues in multi-service networks can be extremely difficult. Typically, technicians will check the health of individual network elements or review a standard set of KPIs when they learn of a problem, but it's rare that an issue stems from a single system and most issues tend to be intermittent. Because issues are hard to isolate, they can even lead to nasty disputes between co-workers. An analytics engine with advanced slicing and dicing capabilities will reveal hidden problems. For example, sporadic voice quality issues could be examined from a variety of angles to determine the root cause. If they are coming from callers in specific area codes, it could indicate a problem with a specific carrier. If they occur after being transferred from a specific voice portal, it could indicate an incorrect setting on a specific server.

Proactively performing this type of network analysis enables companies to predict problems and correct them before they affect customers. When actual failures do occur, multi-dimensional network analytics, coupled with the ability to drill into micro events, significantly reduces time to repair. This can also reduce churn and support costs. To sum it up, the ROI for this type of solution is easily calculated and quickly realized.

Controlling Costs

Expense control is on everyone's agenda these days and understanding end-to-end network traffic patterns can help. For example, when companies originally transitioned to IP communications, they realized a significant reduction in their phone bills. However, as solutions evolve – due to enhancement, acquisition, global expansion or the incorporation of virtual agents – those routing schemes have become more complex. Many times during this process calls are directed off the network and back again. Standard monitoring solutions will show each of these calls as “complete,” giving no indication that a problem occurred or that unnecessary toll charges were incurred. Only a complex analysis of traffic patterns will reveal opportunities to easily cut costs.

Going beyond standard contact center metrics provides greater visibility into individual business unit performance. Rather than reading a report that outlines average call stats, it's far more interesting to understand how long a VIP customer spends talking to a dedicated agent or how often people are requesting certain types of information, such as mortgage rates. Some questions the contact center can gain insight to through analytics include:

- Are these individual groups fully satisfied with self-service options?
- How are different customer segments interacting with the company?
- Do VIPs prefer to make calls where potential customers prefer online chat?
- Are these preferences changing over time?
- How are social media and mobile phones impacting customer behaviors?

When this type of intelligence is correlated, it paints a detailed analysis of network performance, enabling managers to pinpoint infrastructure investments to strategically enhance those experiences.

“Once considered a ‘techy thing,’ networks are now mission-critical for every member of the organization – from the IT manager to the marketing VP to the CEO”

It also helps marketing and sales departments refine their strategies or develop new business models to capitalize on evolving customer needs.

An Effective Commitment

A strong commitment to quality assurance is the only way to ensure a great customer experience and effectively manage technology in today's evolving, multi-service environments. Getting it right with pre-deployment testing ensures that solutions will work as designed when released to customers. Keeping them right with ongoing monitoring ensures quality and availability. Now network analytics provides an opportunity to do what's right with actionable information for smarter decision making throughout the company.

To be effective, companies need a solution designed to handle “big data.” No one wants to hire a team of database analysts and wait three days for an answer. Companies need a network analytics solution that quickly manipulates information and presents it in a meaningful way for specific job functions. Being able to see trends and drill into micro events give troubleshooting technicians an edge. Holistic, end-to-end insight into network performance enables more effective planning, optimization of technology investments and greater control over costs. Executive dashboards and trend reports provide management teams with the intelligence they require to exploit customer needs and find new profit opportunities. With analytics, doing what's right – from the control room to the board room – will give companies a long term advantage in any market. ■

About the Author

Bob Hockman is vice president of product management at Empirix, a provider of service quality assurance solutions for end-to-end comprehensive customer experience management of mobile broadband and IP-based communications systems.



Affording the Upgrade: The New World of Network Intelligence

The ever-increasing demand for “anytime, anywhere” data and how to keep it from shutting you down

BY TERENCE MARTIN BRESLIN AND ANDY HUCKRIDGE

The explosion of mobile applications and IP-based data traffic on mobile devices is fueling the migration to faster data rates and pushing the uptake of 4G technologies. Service providers are busy with migration strategies and upgrading their existing networks as stop-gap measures to allow an all-IP based services platform. Carriers and handset vendors are rolling out application portals as a way of differentiating their offerings while providing better monetization and ARPU. User mobility is pushing the trend for “anywhere, anytime” data technology, while applications are driving the subscriber need.

The march of technology powers much of the above, including more overall data traffic and the move to mobile connectivity/mobile broadband. Applications are becoming pervasive with the subscriber dictating what they want to use, where and how. Operators are continuing to reduce costs by moving to an all-IP core, attempting to reduce network complexity and often outsourcing the management of their networks altogether.

With this much transformation taking place in the network, the migration does not take place over night. In the foreseeable future, network operators still need to support a hybrid network that interconnects various types of existing platforms with next-generation systems and devices. The network may have become “flatter,” but it became more complex at the same time. Therefore, the journey toward a converged all-IP network comes with a whole new set of network performance and management philosophies toward which the IT organizations must adopt and evolve. Real-time monitoring, troubleshooting and provisioning of the network must be implemented

strategically and methodically, as they are driven by the need to maintain and manage the experience of the subscriber. In particular, real-time monitoring of network traffic is proven to be essential in diagnosing and analyzing the performance of the network and the services, and consequently the quality of experience (QoE) of the subscriber.

Legacy Monitoring Schemes – You Can’t Be Real-Time with Legacy Management Tools

Fragmented monitoring approaches exacerbate performance and complexity problems. With the constant push for more efficient connectivity, traditional network monitoring approaches have become inadequate for managing network components on enterprise and service provider



Figure 1

infrastructures. Several major problems have emerged with the growing complexity of data on the network and the accumulation of outdated network monitoring components.

The traditional approach to network monitoring improved the visibility of network performance by placing a host of tools into the network, but unfortunately while the system solves some problems, it introduces others. The biggest challenge faced by IT managers is the inability to access a particular point in a network with multiple tools. This limitation creates a network “blind spot,” making troubleshooting transactions inefficient and difficult to solve. Blind spots occur frequently with the type of overhead management utilized in legacy monitoring schemes. Different sets of tools are scattered across the network in different physical locations, with each vendor’s tools having individual management software inoperable with other vendors. This leads to many blind spots, as the network engineer has limited accessibility to certain points in the network and has to manage an overflow of data. Monitoring costs become increasingly expensive as network management becomes more inefficient. With the increasing cost and reduced ROI, revenues are impacted by the lack of fast and efficient troubleshooting. This fragmented approach to monitoring leads to additional performance and complexity problems.

The Economics of Network Intelligence Optimization – From Cost Center to Profit Center

The key for network operators (telecom, enterprise or government) is to develop a holistic and forward-looking strategy for network monitoring and network management. More important, they must consider carefully the price-performance, diversity, agility and intelligent capabilities of a traffic capture solution. Depending on future requirements, there are a number of macro trends that network operators should be mindful of when planning their network monitoring needs, including “Flattening the Network,” technology development and economics.

The continued explosion of IP will simply accelerate the pace at which legacy systems are displaced by an all-IP network. The “flattening” effect will create more distributed IP components in the network and effectively create more potential points of failure. There will be a broader range of IP services to roll out than anyone can imagine, as a result multiplying the complexity of the network. This creates opportunities for more points of monitoring. The monitoring infrastructure should also be “flat” and flexible across any and all parts of the network.

The Network Intelligence Optimization framework is laying the groundwork for a smart network monitoring infrastructure. In order to sustain the increase in speed, the traffic capture layer must continue to be performed in hardware, where deeper awareness of packets and applications, as well as more dynamic handling of them are necessary.

Network managers must do more with less – tighter budget control while improving service delivery. Traditionally

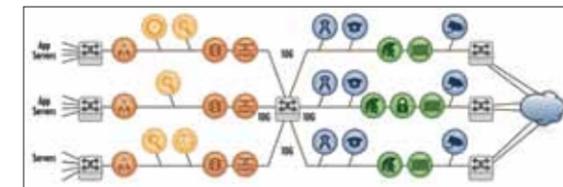


Figure 2

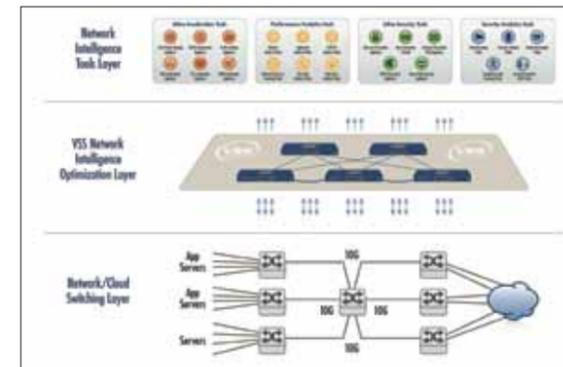


Figure 3

those are conflicting objectives in the IT organization. Conversely, the network monitoring optimization framework allows an organization to migrate from a high initial CAPEX business model to a lower and variable CAPEX model in the network monitoring component of the budget. With less, the network managers can now do more in other areas such as network forensics, lawful intercepts, behavioral analysis, centralizing applications for compliance, etc. In addition, managed service providers (MSP) have become mainstream, where they are focusing on monetization of QoS/QoE, rather than focusing purely on monitoring network elements and packets. The layered approach to network monitoring is fundamental to enabling the business model and differentiation in such network environments. ■

About the Authors

Terence Martin Breslin is CEO of VSS Monitoring. He founded VSS Monitoring in October 2003. His vision of creating a distributed systems architecture to replace the practice of using only standalone TAPs for network traffic capture has changed the practice and potential of network analysis. By providing visibility of any link in even the largest network, VSS Monitoring’s products greatly increase the ROI and productivity of networks and the people who use them.

Andy Huckridge is Director of Marketing at VSS Monitoring. He also serves as an independent Telecom consultant to Network Equipment Vendors (NEMs), Test Equipment Vendors (TEMs), Service Providers focusing in the Test and Measurement industry. Andy has experience in overseeing various international projects in the Telecom / Security and Next-Generation space with leading companies.



Virtually Clearing Cloud Initiatives Through WAN Optimization

Maximizing performance, minimizing costs

BY DR. DAVID HUGHES

There is a storm rising in the midst of enterprise networks today. It builds quickly and overtakes unsuspecting companies as the popularity of virtual initiatives causes data volumes on their networks to swell to proportions too large to access efficiently over existing bandwidth. The challenge then becomes one of optimizing underlying network infrastructure in order to support the increased flow of traffic caused by the data surge.

Application virtualization, cloud computing and Virtual Desktop Infrastructures (VDIs) deliver enormous management and costs savings, but these benefits are offset when application performance hampers end user productivity. This often happens when virtual applications and desktops are delivered across a Wide Area Network (WAN). Optimizing the WAN and improving network stability is therefore paramount to ensuring companies meet their business objectives and take full advantage of the tech-

nology in which they have invested large amounts of money.

Network Performance

Virtual applications are extremely interactive, requiring screen updates and mouse movements to be sent over the WAN using thin-client protocols. While these work fine in a Local Area Network (LAN), there are unique challenges when communicating across a WAN that can have an adverse effect on the performance of these protocols, resulting in slow screen refresh rates and occasional session disconnects. Latency is common when communication takes place over long geographical distances, bandwidth is often limited and costly to provide, and in many environments, like MPLS and cloud, packets are often lost or delivered out of order due to network congestion. When any or all of these issues arise, business continuity initiatives are costly and at risk of failure.

“Virtual applications are extremely interactive, requiring screen updates and mouse movements to be sent over the WAN using thin-client protocols”

The same can be said for end-user applications. Voice, video, file, email and VDIs are all sensitive to bandwidth, distance and network quality, which often lead to poor application performance in remote locations. To accommodate growing data volumes and an ever-increasing need for real-time WAN performance, the average large company upgrades WAN bandwidth approximately every two years. Upgrading bandwidth, however, is both time-consuming and costly, and does not always address application delivery woes brought on by latency, packet loss and other common issues.

For instance, in MPLS, IP VPN and cloud environments, packet delivery issues can increase as throughput increases, which places even more demand on network resources. It's common to see networks with an average packet loss of 0.5 percent reach peaks of five percent, no matter how much bandwidth is available. These issues can lead to excessive re-transmissions, which limits the effective throughput of data transfers across the WAN.

Continuing under the illusion that throwing additional bandwidth at the problem only complicates matters. Instead, enterprises must grasp the importance of tackling the underlying network infrastructure challenges that hamper key business applications.

VDI Victory

Network stability and geographical distances also play a large role in the success of VDIs. As users move farther away from the data center, and as VDI deployments grow in size, it becomes more difficult to deliver a consistent application experience. For one, remote users are often connected via different types of WANs with varying levels of bandwidth, latency and quality. Second, data centers are required to support thousands of simultaneous desktop connections, which presents a unique scalability challenge. Moreover, different VDI applications use different communication protocols, creating an even greater need to stabilize the underlying network infrastructure to ensure maximum VDI performance in all deployment scenarios.

Every Cloud Has a Silver Lining

‘Cloud computing’ is a broad term that is used liberally to mean different things to different organizations. At the highest level, it involves the delivery of hosted services over a shared WAN, such as the Internet. Regardless of the type of service deployed, all cloud computing initiatives have one thing in common – data is centralized, while users are distributed. Understandably, this places increased pressure on the network, making cloud computing susceptible to the same WAN bandwidth, latency and quality challenges that impact other enterprise applications.

To accommodate the increasing volumes of data, organizations can deploy large WAN links between data cen-

REPORTS!

contact
CORINNA MELCON
tel 201-802-3026
corinna@sys-con.com

“By taking a network-centric approach, organizations achieve maximum scalability and the flexibility needed to support all current and emerging applications on the network, and in so doing clear the vulnerability gap that seriously compromises business continuity plans”

ters, which are able to handle hundreds of megabytes or terabytes of data. However, given the price of WAN bandwidth, this can become an extremely expensive component, which rarely solves the true underlying performance problems.

Maximizing Performance, Minimizing Costs

A real-time solution that has the scalability to handle increasing volumes of data traffic – such as WAN optimization – is critical to cloud computing services. It is the only way to overcome bandwidth, latency and WAN quality issues that plague the cloud and business applications in general. Reducing the amount of data sent across the WAN, prioritizing key traffic and eliminating packet retransmissions with WAN optimization

technology has a threefold result: network performance and end-user experience improve drastically, while ongoing telecommunications costs are significantly reduced.

With some business applications costing close to \$10,000 per user to deploy, enterprises find themselves spending considerable portions of their IT budgets rolling them out. It’s common for IT managers to spend a third of their total project investments on upgrades to IT infrastructure in order to support such new applications. Comparatively, a network investment often costs less than one percent of the total cost of deploying a single enterprise application. As a result, WAN optimization provides a rapid return on investment with the added benefit of increased end-user satisfaction.

WAN optimization not only addresses the latency, packet loss and bandwidth challenges that cause virtual applications and VDI to be unresponsive and/or unreliable across the WAN, but it also provides real-time optimization techniques that maximize WAN performance – a fundamental aspect of any successful cloud initiative.

By taking a network-centric approach, organizations achieve maximum scalability and the flexibility needed to support all current and emerging applications on the network, and in so doing clear the vulnerability gap that seriously compromises business continuity plans. ■

About the Author

Dr. David Hughes holds more than 25 patents for networking related inventions. In 2004, he founded Silver Peak Systems, a leader in the Wide Area Network (WAN) Optimization field that improves backup, replication and recovery between data centers, and facilitates branch office server and storage centralization by improving application performance across WANs. As Chief Technology Officer for Silver Peak Systems, he is responsible for the company’s technical direction and product vision. He previously held senior architect positions with Cisco Systems, Stratacom, Blueleaf and Nortel. Dr. Hughes has a PhD in packet network optimization.

Advertiser Index

ADVERTISER	URL	PHONE	PG
eLearnIT	http://www.isug.com/pbtraining		9
PBDJournal	http://pbdj.sys-con.com/		19
Sybase	http://www.sybase.com/powerbuilder		C2 & C4
Sybase	http://www.sybase.com/powerbuilder		20-21
SYS-CON Events	http://events.sys-con.com	201-802-3020	C3

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of PowerBuilder Developer's Journal. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher, Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in PowerBuilder Developer's Journal. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

ALL THINGS OF INTEREST TO THE PB COMMUNITY

Traders to Leverage High Volumes of Data from Multiple Sources in Real Time with Sybase Event Stream Processor

Sybase, Inc., and Headstrong, a global IT consulting firm, have announced that Sybase Event Stream Processor (ESP), the latest release of the Sybase Aleri complex event processing platform, will be fully integrated with Headstrong’s Teevra real-time trade matching and STRIDE retail trading platforms. With Sybase ESP, Headstrong was able to accelerate the development of customer-facing financial applications that leverage high volumes of data from multiple sources in real time. This, in turn, allowed Headstrong to facilitate instantaneous trade verification and validation in one case, and market making and hedging in another case.

As Headstrong began to develop two of its key solutions – Teevra, a real-time trade-matching system, and STRIDE, a trading solution for CFD’s (Contracts for Difference) – the company researched complex event processing (CEP) platforms that could serve as the core engine for specific functionality within these two applications. Headstrong needed a CEP platform that would support the level of transaction rates, throughput and latency that are essential to the performance of STRIDE due to ever increasing volumes across markets.

Sybase ESP empowers software development firms to deliver real-time applications such as continuous risk-assessment and low-latency algorithmic trading. The technology also facilitates the creation and dissemination of highly enriched market data in a fraction of the time that traditional approaches require. Sybase ESP also allows users to obtain intelligence on high-speed streaming data with minimum latency. Sybase ESP enables Continuous Intelligence, letting software developers create solutions that monitor, detect, analyze and respond to events as they occur. This empowers brokers and their customers to make opportune corrective, profitable, competitive and value-add financial decisions.

Sybase ESP, the latest release of the award winning Sybase Aleri CEP platform, consists of a high performance complex event processing engine, the ESP Studio, and a range of integration tools including adapters and APIs. Designed as a new modeling paradigm, Sybase ESP combines drag-and-drop simplicity with a full SQL-based programming language for power users. As a real-time intelligence solution, Sybase ESP comprises a dynamic environment, facilitating the addition of new continuous queries at any time.

Nine Out of 10 Bankers Not Confident Over Stress Testing, While Three Out of Four Struggle to Capture Trading Data, According to New Sybase Survey

Sybase, Inc., has announced results of a survey showing

that more than nine out of 10 capital markets executives lack confidence in bank stress testing, while more than 75 percent are struggling to capture trading system data. These key findings, among others, came from a survey conducted in London at the Sybase Financial Services Executive Summit, which drew senior managers and C-suite risk management professionals from major capital markets firms.

When asked about the reliability and frequency of bank system stress testing, a combined 94 percent of respondents were either not at all confident or only somewhat confident that stress testing has addressed all the important risks to the banking system. Directly related, more than eight out of 10 executives surveyed (84 percent) believe stress testing should be upped to at least every six months. Currently, EU regulators carry out yearly stress tests on the biggest lenders in the region.

Three out of four respondents (76 percent) said they struggle to capture data from trading systems for audit, compliance or future analytical purposes. At the same time, 75 percent believe data latency and regulatory challenges will consume the most significant portion of their firm’s resources, followed by front/middle/back office integration (25 percent). “Merging front and middle office in particular presents numerous challenges, yet they are clearly outweighed by the business benefits. These lines will only continue to blur,” said Stuart Grant, Financial Services Business Development Manager at Sybase.

Basel III’s impact on profitability will range from moderate to significant, according to nearly all (98 percent) of those surveyed, indicating another area of concern added to the Eurozone’s current anxiety over sovereign debt issues plaguing Greece and other countries.

Sybase Does More Big Data Analytics with Native MapReduce and Hadoop Integration

Sybase, Inc., has launched the new version of the Sybase IQ high performance column-based analytics database. The new features of Sybase IQ 15.4 provide advanced techniques to address the new generation of Big Data and unlock critical business insights.

Today’s growing demand for analytics to spot trends, correlate cause and effect, identify patterns, or ask “what if?” questions necessitates new analytics techniques, such as the use of advanced tools and the execution of analytical algorithms close to the data stored in the database. The new release of Sybase IQ expands support for in-database analytics, including a native MapReduce API, Predictive Model Markup Language (PMML) sup-

port, integration with Hadoop, and an expanded library of statistical and data mining algorithms that leverage the power of Sybase IQ PlexQ massively parallel processing (MPP) technology. New APIs enable developers to quickly and safely implement in-database algorithms achieving greater than 10x performance acceleration over existing approaches.

The ability to execute models in the industry standard Predictive Model Markup Language (PMML) format in-database within Sybase IQ allows data scientists to build predictive models in industry standard analytics tools and then automate their execution in Sybase IQ. The PMML capability combined with existing capabilities for text and multimedia analytics provides enterprises with a breadth of available techniques for analyzing Big Data.

Sybase IQ 15.4 delivers APIs to execute proprietary algorithms in-database providing differentiated analytic capabilities with unmatched performance. Using these new APIs developers and data scientists can integrate and execute certified function libraries from Sybase partners and custom-developed models, analytic algorithms, and data mining technologies inside a Sybase IQ database server, easily giving 10x performance gains over traditional approaches. Key capabilities include:

- **Native MapReduce API** using In-Database Analytics and SQL interfaces enabling data scientists to use alternative techniques to analyze massive data sets.
- **Hadoop integration** offering different techniques to integrate Hadoop data and analysis with Sybase IQ.
- **In-database Analytics APIs and tools:** A new Java API and new extensions to the C++ APIs allowing implementation of high performance proprietary or certified ISV algorithms in-database that run 10x faster. In addition, the new in-database analytics simulation environment eases development and testing.
- **Predictive Model Markup Language (PMML) support** through a certified plug-in from Zementis to automate the in-database execution of analytic models created in industry standard data mining tools and other popular predictive workbench products.
- **Updated in-database statistical and data mining library, DBLytx from Fuzzy Logix,** harnesses the power of Native MapReduce API (TPF) to deliver high performance analysis.
- **Better compression of text data** than commonly found in Big Data scenarios.
- **Faster bulk loading** of large data sets through ODBC and JDBC interfaces.

Availability

Sybase IQ 15.4 is generally available, and also introduces a free Express Edition, enabling IT departments, developers and application vendors to easily download and start developing with Sybase IQ. Visit: www.sybase.com/iq.

Sybase 365 Partners with the Qtel Group to Provide Mobile Wallet Solution to Millions Across the Region

Sybase 365, a subsidiary of Sybase, Inc., has announced its strategic alliance with the Qtel Group, a provider of telecommunications services. Sybase 365 will provide its mobile wallet (mWallet) software with comprehensive solutions support, delivering one of the most advanced mobile commerce services in the world.

The newly launched Qtel Mobile Money service, a safe, fast, and easy-to-use mobile wallet offering, will allow Qtel Group customers to use their mobile phone accounts to save and transfer money, top-up their credit and make payments at home and abroad via their mobile devices. The Qtel Group's operations cover markets with a high number of residents and foreign workers, who regularly transfer a proportion of their salaries overseas to their home countries. A large proportion of these workers do not have access to traditional banking facilities and are "under-banked," relying on the services of money transfer kiosks to transport money home.

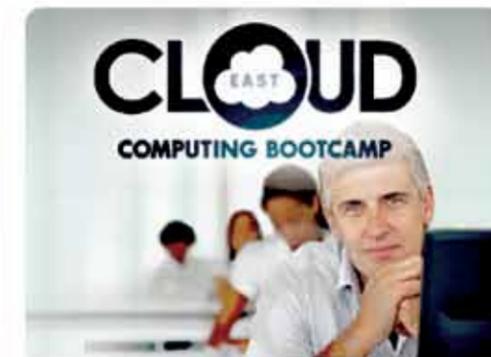
Qtel Mobile Money service enabled by Sybase 365 mobile wallet software will help the Qtel Group support all mobile channels, including SMS, Unstructured Supplementary Service Data (USSD), mobile browser, downloadable client, STK and works on any mobile device, with any currency.

Sybase Dispels the Myths of Big Data with 'Intelligence for Everyone' Guide

Sybase, Inc., has unveiled its new guide on Big Data analytics titled Intelligence for Everyone: Transforming Business Analytics Across the Enterprise. The guide demonstrates through facts and examples that there are tools and methods to make sense of new and massive data sets, service all users at all levels, and analyze many different data types to provide actionable answers.

The Guide presents proof points of how companies have successfully extended analytics deep into their operations across a range of industries. It discusses research studies into the value of analytics and presents case studies on the significant ROI of analytics within enterprises, especially those dealing with growing data sets, expanding user populations, differing data types, rising workloads, and increasing ad hoc queries.

A research study by the University of Texas at Austin revealed that many leading enterprises are not realizing the true value and opportunities of intelligently mining the gluttony of information, or Big Data. Conversely, a recent Sybase infographic illustrates the significant increases to employee productivity and profitable gains in sales that can be achieved with only a 10 percent improvement to data usability and accessibility. ■



June 11-14, 2012 • Javits Center • New York City
November 5-8, 2012 • Santa Clara Convention Center • Santa Clara, CA

EXPERIENCE SYS-CON EVENTS' 2012 CONFERENCES & EXPOS

visit www.events.sys-con.com • tel 201-802-3021





PowerBuilder 12.5 Rocking the .NET world... and so much more!

Celebrate 20 years of blazing speed and simplicity

Sybase® PowerBuilder® is, and has been, the award-winning tool of choice for Windows development teams who want results – not excuses. PowerBuilder 12.5 continues this proud tradition with more features and functionality to support all of your development needs. With two Integrated Development Environments (IDEs), PowerBuilder 12.5 is the second full-fledged .NET release, bringing true power and ease of use to .NET.

But it doesn't stop there. It brings fresh new features to your Win32 applications. We also make it easy to migrate your existing Win32 applications to .NET at a fraction of the time, cost and complication of re-writing. So, the choice is yours – innovate in Win32, rock .NET and WPF, or combine the two. Any way you go, you will have the power to create the most updated, visually appealing business applications – on time and under budget.

Join the revolution. Download a 45-day evaluation copy of PowerBuilder 12.5:
www.sybase.com/powerbuilder

Copyright © 2011 Sybase, Inc. All rights reserved. Unpublished rights reserved under U.S. copyright laws. Sybase and the Sybase logo are trademarks of Sybase, Inc. or its subsidiaries. ® indicates registration in the United States of America. SAP and the SAP logo are the trademarks or registered trademarks of SAP AG in Germany and in several other countries. All other trademarks are the property of their respective owners.

SYBASE®
An **SAP** Company