

Omni 7 Directional



Tim Negris Takes Control

On February 5, 1997, Blyth Software announced that Tim Negris, previously Executive Vice President of Marketing and Development had been appointed as President and CEO. Michael J. Minor stepped down as CEO and Chairman. Stephen R. Lorentzen has resigned his post as President and Chief Operating Officer. Richard J. Hanschen returns to Chairman of the Board.

Negris has over 15 years of experience in software marketing. Most recently, he was Vice President, Sales and Marketing for IBM's multi-billion dollar Software Solutions Division. Prior to IBM, he was Vice President, Server Product Marketing at Oracle Corporation and was one of the primary architects of Oracle7's highly successful market launch.

In this article, Tim Negris, the new Blyth CEO and President, sets out his vision for the future. Tim is a consummate marketing professional. OMNIS developers can now be certain that marketing will be the major change in Blyth over the coming year. Tim presents a fresh, aggressive approach and at last there are real prospects for growth.

Blyth Software – The Next Generation

Think about your favorite dead software companies. They probably all had smart people, good products, and loyal customers. But they died anyway. Now, think about Microsoft, Netscape, Oracle, Powersoft, SAP, and other wildly successful software companies. What do they have that the others did not? There are many possible answers, but the most common ones are likely to be Vision, Value, and Visibility.

Going forward, you can be sure that we, at Blyth Software, are maniacally focused on the fundamentals. We are building a great team, we have a solid roster of new products, and we are listening more closely than ever to our customers. But, we know that this is not enough. Without vision, value and visibility, we too will just become another one of your favorite dead software companies.

I don't intend to let that happen. Nor do I intend for Blyth to merely "recover" from the brutal challenges of recent years only to settle into a tenuous niche until the next paradigm shift. I intend to put the entire industry on notice: We are the next paradigm shift. Bold words? Perhaps. But, as our experienced customers already know, OMNIS is more than just another software development tool. It provides a means for gathering, retrieving, organizing, manipulating, and presenting information that other products cannot duplicate - and it's about to get even better. And, at the heart of OMNIS is technology which will form the basis for a new generation of breakthrough software products, both at Blyth and across the industry.

Allow me now to share with you my vision for the future, our value as a company, and what we are doing to make sure that everyone knows about it.

A compelling vision of a world in which we are inevitable.

Today's software developers must harmonize many different technologies, including SQL, the Web, objects, messaging, transactions, directory services, file systems and more. What is needed is an easy way to blend all of them together into application frameworks which form highly reusable software units whose computational

In This Issue...

- Tim Negris becomes new CEO of Blyth 1
- NeXT 3
- New VP to Lead Marketing Strategy for Providing Access to Mission-Strategic Applications on the Internet 5
- Tips and Tricks 6
- REPCAT: A Desktop Expert System for Interpreting and Validating Laboratory Data 8
- Submissions to OMNIDirectional 9
- The OMNIS Underground Communicate 12
- OMNIS training videos 14
- The OMNIS Advisor 15
- OMNIS Web Rad 16

functions more directly correspond to business and organizational functions.

In the next version of OMNIS, a.k.a. "Prometheus" it will be possible to do exactly that. Not only has this product been endowed with the ability to effectively incorporate each of these discreet technologies into a single program, but its robust object model, coupled with the abstraction of the underlying Dot Notation language, makes framework-oriented programming a reality.

In addition to providing our customers with a powerful new applications development tool, this technology provides Blyth with a way to rapidly extend our own product line, as with the killer new web management tools we are already working on. This same technology has also many uses beyond our own business. Thus, we are going to make it available to tools, platform and middleware vendors for incorporation into their products on an OEM basis. Dot Notation has a unique potential for adoption as the de facto standard for dynamic framework scripting across the industry.

A unique value proposition: Future-proof applications.

The software deployment landscape is getting more and more complicated. Fat clients for robust desktops, thin clients for wide-scale deployment, peer-to-peer for resource sharing, mobile applications for highly distributed systems, and server-based agents for optimized back-end functions. There are good reasons for all of these models and we believe that they will continue to coexist. Saying that one is better than another is like saying a hammer is a better tool than a saw.

OMNIS is the only applications development environment which will enable function delivery in all applications models. Coupled with the powerful additional capabilities of transparent cross-platform deployment, embedded database, team development support tools, and automated software distribution tools, the OMNIS product line lets developers meet constantly changing requirements with future-proof applications.

This little light of mine, I'm gonna let it shine!

To paraphrase a couple of old expressions, "Nobody gets fired for buying a heavily promoted product", and "You can't buy what you can't see." Blyth's recent history proves conclusively that "stealth marketing" doesn't work. Not only does it make it harder to get new customers, but it makes it harder to keep old ones. Our famously loyal installed base is under increasing pressure to defend their choice of a product from an increasingly obscure vendor. Questions about Blyth's longevity abound. Furthermore, our technical packaging, pricing, licensing policies, and so forth, are not keeping pace with fashion. People are confused by our identity. This has got to stop, and it will.

In 1997, Blyth will be substantially increasing its marketing budget and overall promotional efforts. We are resuming regular advertising, trade show exhibition, press and analyst activities, customer success stories, user groups, and joint marketing with our value-added partners. And we are doing it on a world-wide basis.

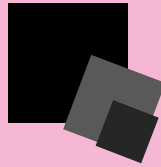
In addition to increasing our promotional activities, we are also working on our overall image. In the coming months you can expect exciting new company and product identities, a user-friendly web site, attention-getting marketing materials, revised packaging and pricing, and new product roll-outs that will not go unnoticed.

As the President and CEO of Blyth Software, I am committed to our success and yours. But I cannot do it without your help. I welcome your suggestions, requests, complaints and feedback.

Tim Negriz
CEO
Blyth Software, Inc.
Foster City, CA, USA
February 19, 1997
<tim_negriz@blyth.com>

Omni Directional is produced by DLA Technology
as a service to users of Blyth Software's OMNIS™

D L A



TECHNOLOGY
MANAGEMENT
CONSULTANTS

A Bi-monthly publication of:

The DLA Group Pty Limited

A.C.N. 003 329 039

Level 3, 35 Clarence Street
Sydney NSW 2000 Australia

Tel: (+61 2) 9262 2255

Fax: (+61 2) 9262 2290

DLA Technology

- ◆ Change Management
- ◆ Database Design
- ◆ Consulting Services
- ◆ Software Development
- ◆ Internet Advertising
- ◆ OMNIS Software distribution and development

DLA Software

- ◆ Solicitor's Companion practice management software for solicitors
- ◆ Counsel's Companion practice management software for barristers
- ◆ Litigation support development and services
- ◆ Databases for marketing, human resources and library management

Email: info@dlagroup.com.au
URL: http://www.dlagroup.com.au
Editor: Stephen Miller
stephen_miller@dlagroup.com.au
Managing Director: David P. Lewis
david_lewis@dlagroup.com.au
Layout: Daniel G. Lewkovitz
daniel_lewkovitz@dlagroup.com.au

OMNIS and OMNIS 7 are Trademarks of Blyth Software Inc. All other product or service names mentioned herein are trademarks of their respective owners.

Copyright (c) The DLA Group Pty Ltd 1996. All rights reserved.

What's NeXT?

Introduction

On December 20, 1996, Apple Computer, Inc., created a stir when it announced at a late Friday evening press conference that it was acquiring NeXT Software for US\$ 400 million in cash and stock, and that it planned to use OpenStep combined with key Apple technologies to create Apple's next generation operating system. For its US\$ 400 million, Apple bought a lot. First, and foremost, it purchased OpenStep with Mach, the most advanced operating system in the world. It also purchased NeXT's WebObjects, a powerful tool for building dynamic Internet web sites using a three-tier client server model. WebObjects is built on top of the base of frameworks and components provided by NeXT's OpenStep and OpenStep Enterprise objects.

And finally, in one of the most interesting twists to this surprising turn of events, Apple brought Steve Jobs, one of Apple's founders, back to the fold as an advisor to Gil Amelio.

What happened to Copland?

Upon hearing that Apple purchased NeXT to acquire OpenStep with Mach as the foundation for its next generation operating system, the obvious question is, "What happened to Copland? I thought that was Apple's next generation operating system?"

The short answer is that Copland tried to be too many things to too many audiences. In the end, it failed to deliver anything for any of its target audiences.

The long answer is that Copland tried to meet the competing goals of providing backward compatibility while delivering powerful new features in a single package. Since these goals were at odds with each other, compromises had to be made. The end result was a series of compromises that ended up creating a product that was neither backwardly compatible, nor capable of delivering an OS for the future.

What happened to "Plan Be"?

Starting in July of last year, up through the announcement of the NeXT-Apple deal, speculation was rife that Apple was planning to buy Be Corp., a start-up founded by former Apple executive Jean-Louis Gassée, that had created a desktop operating system optimized for multimedia.

What happened? Why did Apple not acquire Be and its technology to form the basis of its next generation operating system? In a word, maturity. As Gil Amelio has said numerous times since the NeXT-Apple deal was announced, "NeXT has today, what hopefully Be will deliver in two to five years."

That said, the Be OS is an extremely exciting and interesting OS that will be available on Power Macintosh and PowerPC Platform (aka. CHRP) hardware going forward.

Why NeXT?

In a nutshell, Apple chose NeXT's OpenStep with Mach as the foundation for its next generation operating system, because it provides the richest foundation and most complementary platform on which to build Apple's next generation operating system.

The richness of OpenStep is testified to by NeXT's customers. Companies and organizations like the United States' Central Intelligence Agency (CIA) and National Security Agency (NSA); J.P. Morgan; AT&T Wireless; CERN — where Tim Berners-Lee used OpenStep to create the first World Wide Web client and server in just two months; Merrill Lynch; Fannie Mae — the largest purchaser of home mortgages and underwriter of mortgage backed securities in the United States; and many others, have found that using OpenStep they can create better, more usable, and more functional software an order of magnitude faster than on any other platform, including Mac OS, Windows 95 and Windows NT.

Further, from the list of companies above, it is clear that there has been high level business acceptance of the NeXT technology. Apple has not only purchased an excellent operating system base, we have also demonstrated our commitment to these FORTUNE 100 corporations and will now be able to refocus on the critical business sector.

Introducing Rhapsody...

Like all upstanding Apple projects, Apple's next generation operating system has been given a code name: Rhapsody.

The goal of Rhapsody is to fuse the best from OpenStep with Mach — the OS Kernel, the OpenStep frameworks — and Apple's cutting edge graphic and multimedia technologies — QuickDraw 3D, QuickDraw GX Line Layout and Fonts, QuickTime, and QuickTime VR, OpenDoc, and other cutting edge technologies, to create an operating system that leapfrogs the competition and reintroduces excitement to the personal computing industry by making it possible to create applications that could not previously be created.

Apple's goals with Rhapsody are:

High performance & stability

For years Macintosh hardware has been prevented from delivering its full performance to customers due to the limitations of the Mac OS. With the Mach microkernel and the OpenStep frameworks, Apple will deliver a modern operating systems that provides and takes advantage of the facilities for preemptive multitasking, multithreading, memory protection, and symmetric multiprocessing. Best of class Internet support

Today, the Mac OS provides the personal computer industries easiest to use Internet access and delivery features. Starting with the features that make the Mac OS the easiest to use Internet client and server, and building on the operating system features supplied by Rhapsody, as well as Rhapsody's built in support for Java, Apple will deliver the easiest to use and most powerful Internet client, server, and content creation platform available.

State-of-art multimedia

Apple's industry leading graphics and multimedia technologies — QuickDraw GX line layout and fonts, QuickDraw 3D, QuickTime, and QuickTime VR — will be incorporated into Rhapsody.

Ease of use

The Mac OS is still the definition of ease of use. Rhapsody will preserve and build upon the "look and feel" of today's Mac OS. Building upon the foundation provided by the Mac OS "look and feel" Rhapsody will add improvements from Apple Labs and NeXT to produce a user experience that while familiar, takes a leap forward.

Wicked fast development

The object-oriented development environment provided by OpenStep is renowned for its ease of development and incredible developer productivity. OpenStep has found its greatest acceptance from in-house developers working to create mission critical applications quickly and easily. An acceptance that is based on the order of magnitude greater developer productivity offered by OpenStep. This experience will be enhanced and improved as OpenStep is evolved to create “The Yellow Box” through the integration of key Apple technologies into the object oriented APIs supplied by OpenStep.

Compatibility

Unlike Copland, Rhapsody will offer nearly 100% compatibility to today's Mac OS software. It will achieve this by creating a Mac OS virtual machine (aka. “The Blue Box”) into which will be loaded an image of the Mac OS ROM and System Software. Because you will for all intents and purposes be running your software on a Mac (though a virtual one), you should be compatible unless you directly read or write to hardware.

The Rhapsody Architecture . . .

Power Macintosh, PowerPC Platform (aka. CHRP) Hardware — Rhapsody will run on Power Macintosh and PowerPC Platform (aka. CHRP or Common Hardware Reference Platform) hardware. While the developer releases and even the Premier Release of Rhapsody will be limited in the hardware on which they run, Apple's goal is to support all currently shipping PowerPC processor-based Macintosh systems including Performas, PowerBooks, and Power Macintosh systems. As Apple is able, it will work to extend this support to include any system that has a PowerPC logic board (i.e. excluding systems upgrade via a processor upgrade card).

Core OS —

Apple recently announced that Rhapsody will for a variety of reasons — both technical and business — use Mach as its microkernel. Apple is still in the process of designing the I/O architecture, file systems, networking, etc. that will make up the Rhapsody Core OS.

The Mach microkernel is a light weight OS kernel that provides the following four services:

1. Process management and scheduling
2. Multithreading
3. Memory protection
4. Messaging

Unlike many other kernels, Mach does not directly provide the file system, networking, and other services that are often consider duties of the kernel. Rather, these services are provided by a number “servers” which are loaded on top of the Mach kernel. This will allow Apple to deliver different versions of Rhapsody tuned to particular applications. For example, you can easily imagine an enterprise server version of Rhapsody that includes an ORB (Object Request Broker). Driven by the desire to get a release of Rhapsody into developers' hands as soon as possible, Apple will initially ship on top of Mach 2.5 with NeXT enhancements. This is the version currently used by OpenStep with Mach. In subsequent developer and commercial releases Apple will evolve the microkernel and the other components of the Core OS.

Mac OS Compatibility (aka. “The Blue Box”)

Mac OS Compatibility — which is called “The Blue Box” — will be achieved by creating a Mac OS virtual machine that runs as a single Mach OS process.

The Mac OS virtual machine will be created by loading an image of the Mac OS ROM and current version of the system software into a single virtual address space. This will allow existing Mac OS software — applications, system extensions, control panels, etc. — to run as long as they do not directly read or write hardware. This will provide a much higher level of compatibility than was planned for Copland. While the Blue Box will be memory protected from the rest of the Rhapsody system, and vice versa, applications running within the Blue Box will not be protected from one another. This means that a single misbehaving application can bring down the entire Blue Box. The upside, is that your entire system can't be brought down by an errant application, and that a reboot is only a double-click away!

OpenStep-based APIs (aka. “The Yellow Box”)

From the application software developer's point-of-view, the most exciting part of Rhapsody is the Yellow Box, a collection of object oriented APIs.

Built on the foundation of the OpenStep Dynamic Application System — which provides a dynamic runtime vs. the static runtime provided by C++ — OpenStep provides a dynamic environment in which objects can be bound, loaded, and created on the fly. This allows for great flexibility in the design, development, and deployment of applications based on OpenStep.

OpenStep provides a a rich set of frameworks for building applications. If you have used a C++ based framework like MFC (Microsoft Foundation Classes), MacApp, TCL (Think Class Library), or PowerPlant, you have had only a slight taste of what OpenStep will do for you.

It is not overreaching to say that OpenStep provides an order of magnitude (yes, 10 times) improvement in developer productivity. If you do not believe me, I urge to check out the suite of applications that was created by Lighthouse Design <<http://www.lighthouse.com>>. Each of these applications — numbering 15 in total — rivals its best competitor on the Mac OS and Windows, yet the total number of employees in engineering at Lighthouse Design has never been greater than 15! Think of it, 15 applications created by 15 developers!

The Yellow Box will not just be a rehash of today's OpenStep APIs (though a tasty hash that would be). Rather, Apple will be integrating key Apple technologies from the areas of graphics and multimedia — including QuickDraw 3D, QuickDraw GX Type and Font, QuickTime, and QuickTime VR ; component software — OpenDoc; and others. Our goal is to take OpenStep — easily the most powerful and flexible development platform available — and make it even more powerful, flexible, and feature filled.

Java VM

Java — the language and the runtime system — are integral parts of the Rhapsody architecture. This integration of Java into Rhapsody is made possible due to the commonalities between the Objective-C

runtime — which forms the basis of the OpenStep Dynamic Application System — and the Java runtime.

What this means for you as a developer is that you will be able to freely intermix your use of Java classes and Objective-C objects, with Java classes subclassing from Objective-C objects and vice versa.

Advanced Mac Look and Feel

A key point to get across is that Rhapsody will have the Mac OS user experience! Drawing on Apple — drawing from the Copland project and Apple Labs — and NeXT's — as embodied in OpenStep — work in user interface, Apple is creating a user experience that is distinctly Mac-like, but also cutting edge. If you are a Mac user, you will not be on unfamiliar ground when using Rhapsody. Rather, you will be experiencing the next generation of the Mac user experience.

Going Cross-Platform with OpenStep

Currently, OpenStep is a cross-platform development framework that supports OpenStep with Mach, Windows NT, and Solaris. Apple will continue to support OpenStep as a cross-platform development framework, so that our developers can write once and reap the benefits of deploying on multiple platforms.

When does it ship it?

Rhapsody will first appear as a developer release sometime towards the middle of this year. This first developer release will simply be an implementation of the Yellow Box and the Mach microkernel running on select Power Macintosh hardware. It is important to note that it will not include the Blue Box.

The developer release will be followed by the premier release of Rhapsody, which will be targeted to early adopters, who want to get experience with the Rhapsody, but do not demand a fully implemented or integrated Blue Box. The premier release of Rhapsody will ship towards the end of this year/the beginning of next year.

The final, unified release of Rhapsody, with a fully implemented Blue Box, will ship towards the middle of next year. Apple is working with software developers to make sure that when Rhapsody ships, that they are ready with exciting applications that take full advantage of the functionality and features provided by this new operating system.

Rhapsody & OMNIS?

As OMNIS developers, you are probably wondering what the future for OMNIS and its associated product is on Rhapsody.

As one of the tools evangelists at Apple Computer, it is my job to work with key tool vendors to make sure that they support Apple's business and technical directions. I work with a portfolio of accounts that includes such heavy weights of the tools world as Metrowerks, OmniSoft, ACI, Aladdin, MindVision, your own DLA Group, and others. And while I have nothing specific to announce concerning OMNIS for Rhapsody, my goal is to make sure that they support Rhapsody across the board.

So, how can I get started?

The best way to get started preparing yourself for Rhapsody development is to immerse yourself in the currently existing OpenStep documentation, which is available in PDF format and can be downloaded from the NeXT Web Site at <<http://www.next.com>>.

I recommend that you start with "Discovering OpenStep" and move from there into the other documents that are available.

Questions?

I am sure that each and everyone of you has questions about Rhapsody and Apple's strategy going forward that I have not answered in this article. For that reason, I am offering to take your questions and answer them. So, please send your questions to me in care of the DLA Group at <<mailto:jordan@dlagroup.com.au>>, and I will answer them in future issues of *OmniDirectional*.

By *Jordan J. Dea-Mattson*

Senior Evangelist

Tools & Utilities

Apple Computer, Inc.

Jordan Dea-Mattson (formerly known as Jordan Mattson in his pre-nuptial state) is a 10 year veteran of Apple Computer's developer relations programs. During his tour of duty with Apple, he served in the PowerPC campaign, where he evangelised Metrowerks. He was recently hoodwinked into re-enlisting and is now working on Rhapsody development tools. A strong convert to the next (or is that NeXT) cause, he is very excited about the opportunities that Rhapsody holds. Jordan lives in Mountain View, CA with his wife Wendy and numerous stuffed animals.

New VP to Lead Marketing Strategy for Providing Access to Mission-Strategic Applications on the Internet

FOSTER CITY, Calif., Feb. 5 /PRNewswire/ — Blyth Holdings, Inc. announced today that Pat McEntee has joined the company as Vice President of Marketing. McEntee joins industry heavyweights Tim Negris, President and CEO, and Bob West, Vice President and General Manager Communications Industry, who have joined the company in the last two months to aggressively develop and market OMNIS products. Having led successful marketing efforts at Oracle, Ikonix Inc. and Interactive Digital Solutions, an AT&T/Silicon Graphics venture, McEntee is known for his aggressive marketing strategies and proven expertise in promoting high-end development tools and Internet products.

"Pat's broad experience in developing and executing hard-hitting marketing programs will be invaluable to Blyth as we launch a new generation, and support existing OMNIS products," said Tim Negris, President and CEO of Blyth Software.



Pat McEntee

Tips and Tricks

Omnispic.Ibr Alternative Available

It's basic, but does the job in a clear and concise manner:

- Create, Edit & Delete Groups.
- Add, Edit & Delete Pictures.
- Move Pictures easily between Groups.
- Export Groups & Pictures in a single ODT file.
- Import them back in when needed in a controlled manner.
- Preview Import files.
- A tiny 50KB.

<http://www.gla.ac.uk/Acad/PGMed/staff/gss/goodies.html>

Remember!

That you can improve startup times for OMNIS on the Mac by removing unused extensions and DAMS.

Smartway Win95 and NT Patch Upgrade

If you are looking for ways to improve usability of OMNIS 7.3.2 applications (16bits) under Windows NT 3.51 and NT 4.0, there is a new release of the Win95 patch with specific NT fixes.

Alain Stouder

Smartway SA, Software & Services

Liotard 34, 1202 Geneva, Switzerland

Phone: +41(0)22 344-86-50

Fax: +41(0)22 344-86-67

info@smartway.ch

http://www.smartway.ch

\$statusbar/\$hasstatusbar

The correct syntax for adding/removing status bars to/from an open window is:

Calculate #F as

```
$winds.<WINDOWNAME>.$statusbar.$hasstatusbar.$assign(kTrue/  
kFalse)
```

The correct syntax for adding/removing panes is:

Calculate #F as

```
$winds.<WINDOWNAME>.$statusbar.$panes.$add(<# of  
panes>)
```

Note Page 39 of the Installation and Update manual is not very clear about this.

Courtesy: Blyth Technical Support

AppleEvents and the DoScript Command

The following is the basis for using AppleScript to communicate with a wide variety of applications.

First, due to the paucity of detailed documentation on exactly how all this really works at a low-level, it seems prudent to me to execute the 'Enable Receiving of Apple Events' command. This allows OMNIS to receive non-compulsory events from any application. It is

reversible so just enclose at the beginning of your 'Conversation' routine. Though you may not intend to receive events from remote apps, it is not clear to me that OMNIS and other applications don't carry on some low-level conversation which this command facilitates. I consider it a cheap defensive measure.

You designate who you want to have a conversation with by using the 'Set Recipient' command. I have found OMNIS to be a bit touchy with the names even though the docs (Ref 6:339-356 & DGuide 8:49-58) state OMNIS can do an 'inexact' match on application names. This is that name shown in the application menu so it is subject to user naming. This command sets the flag so you can test to make sure the recipient is set. The 'Prompt for Event Recipient' will allow you to send events to apps on Mac's elsewhere on the network (yes even in different zones!! There are other ways to do this but that is getting pretty far afield {It might even be that, using TCP/IP, with just a little work even IP's of Mac's elsewhere on the Internet could work..hmm }). There are ways of using AppleScript itself to get the names from the Application menu.

Every application has its own set of commands/objects that it can recognize in AppleScript. Any Applescript script must use these commands and objects when "talking" to that application. To determine what these are use the Apple supplied Script Editor (or another tool) and "Open Dictionary". If you open Navigator 3.0's dictionary you will see the 'OpenURL' command, among others. This is the command you want to use. The syntax is: OpenURL "www.blyth.com/".

Pop quiz time. Why won't the following code work?

Begin reversible block

Enable receiving of Apple events

Set event recipient (Navigator 3.0)

End reversible block

```
Send Core Event {Do Script ("OpenURL "www.blyth.com/  
")}
```

;opens the URL in top browser window

Answer: Check that Navigator Dictionary again. You don't see the Do Script command in Navigator's dictionary do you? This set of OMNIS code would send the DoScript command to Navigator who would return a -1708 error, command not understood.

Some applications DO recognize the DoScript command. When AppleScript sends the DoScript command to the application it must use the syntax of the internal scripting language of the application (HyperTalk, WordBasic, etc.). When you send OMNIS a DoScript command, the script you use is what you get when you convert to freetext entry in the procedure editor. Navigator does not have an internal scripting language, hence no recognition of the Do Script command.

This poses a central problem for using OMNIS and AppleScript. How do you send commands to applications if they do not recognize the Do Script command? OMNIS only provides Database events (not many app's use these); Finder events (great for Finder but not anything else); and Core events (Set Data, Do Script, Open Document, Print Document).

Answer: Use an intermediary script dispatcher. Essentially you create a target application which DOES understand the Do Script command and sends the script to the desired application as AppleScript commands. Thankfully, you can create this intermediary application in just a few seconds and it only takes 200K of RAM. Here's how:

Below is the text to an AppleScript which does just what you need. Copy the text into a new Script. Then Save it as an application which has both Stay Open and Never Show Startup Screen turned ON. You want Stay Open ON because you will launch this application from inside OMNIS and then you want to carry on conversations with it. Don't Show Startup Screen is ON because you will launch it discreetly and Quit it discreetly so the user may never need to know it is there. I call this application "OmniScript".

```
on run
when the application is launched
tell application "Finder"
;requires either System 7.5 or Scriptable Finder extension
set ListOfOmnis7v3Names to (every process whose
creator type is "O7$3") as list
if number of items in ListOfOmnis7v3Names = 1 then
set frontmost of every process whose creator type is
"O7$3" to true
;If Omnis7 v3.x is running bring it to front.
else
quit
;just in case the user double click's the app. (Also if 2
instances of OMNIS running...)
end if
end tell
end run
```

— this handles the Do Script events sent later on by OMNIS

```
on do script TextOfScriptFromOmnis
run script TextOfScriptFromOmnis in "AppleScript"
end
do script
```

— this requires the "Run Script" scripting addition. — this addition is part of the standard package from Apple—scripting additions are found in the Scripting Additions folder in the Extensions Folder—NOTE: dropping scripting additions on the System Folder doesn't work

As I mentioned at the outset, this is a generalized solution to communicating with any AppleScript aware application. When launched from inside OMNIS 'Launch program (do not quit OMNIS), this will immediately bring OMNIS back to front. Now you construct the desired script in a text variable in OMNIS.

Any AppleScript has a target application which is receiving the Script command. This is set by using the "Tell application" command. The AppleScript is: tell application "Navigator 3.0"—bring to front activate OpenURL "www.blyth.com/"end tell

The effect is achieved in OMNIS as follows:

```
Begin reversible block
Enable receiving of Apple events
Set event recipient (OmniScript)
Calculate ICR as chr(13)
Calculate fvScript as con('tell application "Navigator
3.0"',ICR,'activate',ICR,'OpenURL "www.blyth.com/
"',ICR,'end tell')
```

```
End reversible block
Send Core event {Do Script (fvScript)}
If (#ERRCODE=0)
OK message ([#ERRCODE] [#ERRTEXT])
End If
```

NOTE: Navigator and OmniScript should be running at the time this is issued.

Hopefully it is fairly obvious how to extend this to obtain quite sophisticated interactions. For example, I have developed a report writer which writes actual Excel worksheets...

Hope this long-windedness proves useful to some! Any AppleScript work out there?

*Contributed by: Steve Cronin
SCronin@AOL.com*

The Deployment Tool

Designating Directories

-how to designate a directory?

-does the deployment tool need to create the directory? if so, how?

You use the Set(destination,x) command.

```
Set(destination,"%D") ;; sets destination to what the user
picked
Set(destination,"%S") ;; sets destination to system/
windows folder
Set(destination,"C:\DIR") ;; sets destination to "C:\DIR"
Set(destination,"%S\DIR") ;; sets destination to the
directory "DIR" inside the system/windows folder.
Set(destination,"%D\DIR");; sets destination to the
directory "DIR" inside the directory the user picked folder,
etc.
```

Alternatively for some commands you can set the destination to nothing, and have the full destination path and file name in the Unpack or Copy command.

```
Set(destination,"")
```

In addition there are a set of installer variables from which you can get paths:

```
%_INSTALLERPATH ;; the full path from were the installer
is launched
%_DESTINATIONPATH ;; the full path the user has picked
%_SYSTEMPATH ;; the full path to the system folder or
windows directory
```

If you always wanted to install in the root of the system drive you could get the system path and extract the drive name. See enclosed sample script on how to extract the drive name. Use the supplied installer. It includes the debugger so you can step through it.

However if you can't work out what the destination should be from the information above, there is nothing in the installer to help you with that.

Continued on page 10

REPCAT: A Desktop Expert System for Interpreting and Validating Laboratory Data for Pheochromocytoma Diagnosis.

Another Australian Success story using the Database Application OMNIS 7™

In this heavily edited and abridged article, Dr George Smythe and Catherine Drew set out the utility of OMNIS 7 as their Expert System. An outline of their methodologies is set out below. The full article may be found in *Clinical Chemistry* 43, No. 1, 1997.

Non standard abbreviations used: PHEO, pheochromocytoma; GC-MS, gas chromatography-mass spectrometry; LIS, Laboratory information system; PC, personal computer; DHPG, 3,4-dihydrophynel glycol; DOPAC, 3,4-dihydrophynelacetic acid; NE, norepinephrine; and EP, epinephrine.

Abstract

To aid in the validation and interpretive reporting of results from multianalyte diagnostic testing for pheochromocytoma, we have developed a desk-top PC-based laboratory expert system (REPCAT). REPCAT utilises a commercial database application to run procedures which assess analytical and clinical data relating to patient urine or plasma samples. REPCAT was used to evaluate the raw data from over 4,000 24-hour urine samples submitted to our laboratory for testing for the presence of pheochromocytoma. REPCAT performed equivalently to an expert pathologist in assessing the presence and class of pheochromocytoma (epinephrine, norepinephrine or mixed secretor). No false negatives were generated and it assigned a correct interpretation (based on subsequent clinical and biochemical investigation) for each of the primary diagnostic samples from these patients.



Dr. George Smythe

Indexing Terms: *Pheochromocytoma, expert systems, computer-aided diagnosis*

Introduction

With increased automation and productivity in the modern pathology laboratory, the process of validating and interpreting laboratory data can be greatly assisted by the use of some form of computerised decision support system. Expert systems are playing an increasingly important role in this area by enabling the generation of interpretive pathology laboratory reports which can be used in clinical decision support (1,2). In many diagnostic investigations the results from several analytes, as well as specific clinical information and knowledge of the therapeutic regimes, are required for an accurate assessment of the patient's status.

Such an example is the biochemical investigation of pheochromocytoma (PHEO), an uncommon tumour which is easily cured but potentially fatal if diagnosis is not made. Recent investigations have indicated that the diagnosis of PHEO in all patients with the tumour requires a minimum of five analytes to be assessed, both individually, and in relation to one another (3,4). As a tertiary referral centre for PHEO diagnosis we have accumulated a base of over 8,000 'non-tumour' patients and some 75 patients with PHEO (3,4,5). These data have been maintained on a Macintosh™ personal computer using database applications developed using the database development software products OMNIS 5™ and OMNIS 7™ (Blyth Software Inc, Foster City, CA). As well as maintaining the clinical and analytical data, these applications have been used to calculate patient results (and QC) from raw data derived from a GC/MS system used in the analysis of urinary and plasma catecholamines and metabolites. More recently we developed an expert system within the database application to interpret individual patient data in relation to reference intervals from the whole non-tumour population in the database (6). Modern PC-based database systems with lookup file capability provide the necessary framework for developing expert systems within a database and we took advantage of these to develop REPCAT. In routine use REPCAT is a fast, easily maintained, application which examines eleven analytes and specific data from the sample request for biochemical investigation of PHEO and produces an interpretive report on the results which is equivalent to that from an expert pathologist.

Materials and Methods – Equipment

REPCAT is an application developed using OMNIS 7™ (Blyth Software Inc, Foster City, CA, USA/DLA Technology, Sydney Australia) which operates on Macintosh™ computers or IBM-compatibles under Windows™. In routine laboratory use, REPCAT runs on a Macintosh LC575 configured with 8Mb of RAM which is linked to a laboratory information system. Development work was carried out on a Macintosh Centris 650 computer. Patient sample records (8300 in the working database) are maintained in an OMNIS 7 data file and currently require 6Mb of hard disk drive space.

Patient database

The full database contains over 8000 records of patient urine and plasma samples referred to us over the past ten years for query PHEO. These records contain 65 datafields in the main file and include patient and sample details, clinical comments, and the analytical results for up to eleven analytes. The analytes currently included are the primary catecholamines norepinephrine, epinephrine,

dopamine, their α -methyl analogues (present if the patient was receiving methyl dopa therapy), their primary neuronal metabolites DHPG and DOPAC, the O-methyl metabolites normetanephrine and metanephrine, and creatinine.

Before an analytical run, patient data are imported as text files into the OMNIS 7 database from the hospital mainframe laboratory information system and clinical notes are keyed in from the sample request. After sample analysis, raw data are imported into the database from the GC/MS and HPLC instruments used for the assays and final analyte concentrations are automatically calculated.

Structure of REPCAT

In OMNIS 7, procedures are built from a list of user-available commands to carry out calculations and/or logical processing of any record or records within the database. Conditionals can be applied in this way to test the result applying to the current (sample) record against a rule in the knowledge base. 'Inferencing' was readily accomplished via OMNIS 7 procedures which we incorporated into the REPCAT application and that extra features found in an *inference engine* or expert system 'shell' (Z). were unnecessary. An 'expert report' by REPCAT is routinely initiated via a procedure call made at the end of data entry; the task of this procedure is to "seive" the sample data through a series of procedures to compare the sample results with data from the knowledge base.

Discussion

REPCAT has been in full-time use in our laboratory for five years. It has proven a fast, highly functional tool, providing the performance of an experienced expert in the domain of investigation of PHEO. It has been shown to be 100% sensitive in predicting a diagnosis of PHEO, as well as the PHEO type. It is at least as specific as an expert can be with current knowledge and significantly more specific than if diagnoses were simply based on one class of analyte (ie, catecholamines or metanephrines). This effectiveness of REPCAT lies in the nature of the domain under consideration; the knowledge is all laboratory based and none is 'missing' (not entered).

Aside from its accuracy and efficiency as an expert system, we believe that by its incorporation into a database application, REPCAT represents a simple approach to practical expert system development in a relatively narrow domain. The method is similar to the concept of a 'data dictionary' (9) which provided a theoretical general basis for systems in which the knowledgebase was a maintainable 'look-up' file. This approach should be applicable in any domain in which the knowledge base is essentially complete and test results are accurate. If such is the case then even complex diagnostic decision making can be eased because a large numbers of variables and combinations can be incorporated into the look-up file. All operations are carried out within the database itself.

REPCAT is in routine use in our laboratory and has eased a significant burden in the area of validating and reporting biochemical data for investigation of patients for pheochromocytoma. Importantly, it empowers laboratory staff to make informed decisions about the further handling of results and reports. It has proven a valuable educative tool and has resulted in the laboratory staff having a greater appreciation of clinical outcome for the patient.

We believe that REPCAT, or other applications derived from database tools such as OMNIS 7, has the potential for use in expert system development in other domains in clinical chemistry.

References

1. Valdiguie P, Rogari E, Philippe H. VALAB: Expert system for validation of biochemical data. *Clin Chem* 1992; 38:83-87.
2. Edwards G, Compton P, Malor R, Srinivasan A, Lazarus L. PEIRS; A pathologist-maintained expert system for the interpretation of chemical pathology reports. *Pathology* 1993; 25:27-34.
3. Duncan MW, Compton PJ, Lazarus L, Smythe GA. Measurement of norepinephrine and 3,4-dihydroxyphenylglycol in urine and plasma for the diagnosis of pheochromocytoma. *New Engl J Med* 1988; 319:136-142.
4. Graham PE, Smythe GA, Edwards GA, Lazarus L. Laboratory diagnosis of phaeochromocytoma: which analytes should we measure? *Ann Clin Biochem* 1993; 30:129-34.
5. Smythe GA, Edwards G, Graham P, Lazarus L. Biochemical Diagnosis of pheochromocytoma by simultaneous measurement of urinary excretion of epinephrine and norepinephrine. *Clin Chem* 1992; 38:486-492.
6. Smythe G, Drew C. A desktop system for handling and reporting laboratory data based on OMNIS 5: application to phaeochromocytoma diagnosis [Abstract]. *Clin Biochem Revs* 1992; 13:104.
7. Winkel P. The application of expert systems in the clinical laboratory. *Clin Chem* 1989; 35:1595-1600.
8. Edwards GA, Smythe GA, Graham PE, Lazarus L. The impact of recent advances in diagnostic technology on the clinical presentation of phaeochromocytoma. *Med J Aust* 1992; 156:153-157.
9. Jansen R, Compton P. The knowledge dictionary: a data dictionary approach to the maintenance of expert systems. *Knowledge based systems* 1989; 2:14-26.

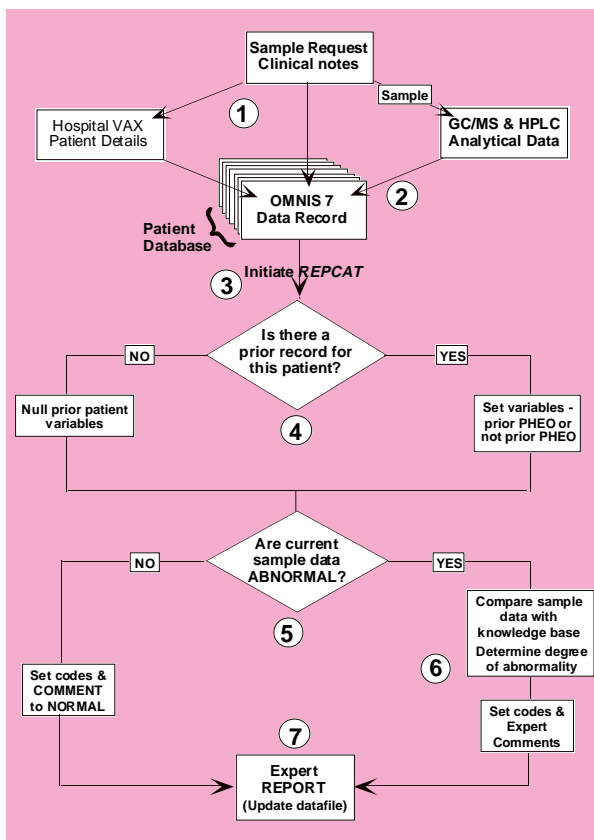
Continued on page 10

Submissions to OMNIDirectional

If you have an article, case study, letter to the editor or hot tip, why not submit it to OMNIDirectional and share it with other readers.

Articles may be submitted via snail mail, preferably by email to omnidirectional@dlagroup.com.au Please ask about submission of photos, screen shots and so on. Writers of published articles may choose from our range of OMNIS RAD t-shirts, mousemats and assorted other goodies!

Figure 1. The steps in generating an expert report by REPCAT



(1) Patient sample details are imported into the database; clinical notes are entered manually; (2) Raw analytical data for each analyte are imported from GC/MS and HPLC instruments to the database. Actual results are calculated by REPCAT (OMNIS 7); (3) REPCAT runs automatically when data entry is complete but can be initiated at any time by a pushbutton control; (4) REPCAT begins by searching the whole patient database for a previous record from the same patient.; (5) By referring to the knowledge base the sample data are first checked for complete normalcy by a single rule which tests all analytes, urine volume and creatinine; (6) Any abnormality in the sample data initiates comparison of the sample data with the expert rule base through a hierarchical series of procedures. These procedures allow the data to pass through the conditional rules until there is a correspondence and a rule is triggered and the associated comment and codes are written to the sample data record; (7) A custom printed report is generated for the current assay. Interpretative comments applying to abnormal sample data are printed along with the analytical results.

George A Smythe and Cathryn M Drew
Biomedical Mass Spectrometry Unit,
University of New South Wales,
Sydney, NSW, 2052 Australia
Tel: + 61-2-9385-1994
Fax: + 61-2-9662-4469
g.smythe@unsw.edu.au

Windows 95 Issues

The deployment tool only works reliably with DOS compatible file and directory structures.

Courtesy Blyth Technical Support

Adding Vertical Row Dividers to Tables

Manually :

-open window with table in design mode

-double click on side vertical scrolling bar (or somewhere else convenient) of table to open the field tools box (or whatever it is called) for that table

-using shift-option with left hand, use right hand on mouse to drag a horizontal divider from the top most line of the table. Drag this just far enough to see it in the middle of your cell height.

-Click on the little 'R' to select the row line. Change its line style to none (\$tablinestyle=0) The row line disappears from sight

-Now drag your new line to the same position as the 'R'. Change its line style to whatever you want

-Reselect your table and header sections from the field tools box

-Save changes and open window to view..this should do what you want

Using dot notation:

To add a horizontal divider to your table row, at the location of the row section line; and, at the same time, change the linestyle of your row section line to 16 (none, invisible), use the following code: (this code can be pasted into a menu line of your lib..of course, substitute your window name and table \$ident number..use att inspector)

-to add horizontal divider to a table's row

```

Local variable LVREF1 (Item reference)
Local variable LVREF2 (Item reference)
Set reference LVREF1 to
$root.$windows.W_Telephone_Log_cp.$objs.1079
If LVREF1
; makes row section line style of row section line none (invisible)
Calculate #F as LVREF1.$tablinestyle.$assign(16)
End If
; adds a line which should be movable by user at position of R (bottom of row)
Set reference LVREF2 to
$root.$windows.W_Telephone_Log_cp.$objs.1079.$horzdividers.$add (LVREF1.$tabrheight)
If LVREF2
; change style of horizontal divider to a thin solid line
Calculate #F as LVREF2.$linestyle.$assign(1)
End If
  
```

The following code will reverse the above notational action. (pasted into and accessed from another menu line)

```
Removes horizontal dividers from same table and make
row section line visible again
Local variable LVREF1 (Item reference)
Set reference LVREF1 to
$root.$windows.W_Telephone_Log_cp.$objs.1079.$horzdividers
If LVREF1
  Local variable LVREF2 (Item reference)
  Set reference LVREF2 to
  $root.$windows.W_Telephone_Log_cp.$objs.1079.$horzdividers.$first
  While LVREF2
    Calculate %NO1 as LVREF1.$remove(LVREF2)
    Set reference LVREF2 to
    $root.$windows.W_Telephone_Log_cp.$objs.1079.$horzdividers.$first
  End While
End If
; change bottom row section line to visible again
Set reference LVREF1 to
$root.$windows.W_Telephone_Log_cp.$objs.1079
If LVREF1
  Calculate #F as LVREF1.$tablinestyle.$assign(1)
End If
```

note., you have to reopen the window to see changes as they are notationally made to the closed window formats (\$windows) instead of the open window formats (\$winds).

A quick tool which works like the Attribute Inspector except you fire it from a menu line ...

```
Local variable the_ref (Item reference)
Calculate $root.$modes.$fixedcursor as kTrue
Calculate $root.$modes.$ccursor as kcursExamine
While mouseup()
End While
Calculate $root.$modes.$fixedcursor as kFalse
Calculate $root.$modes.$ccursor as kcursCross
Set reference the_ref to mouseover(1)
Local variable ItemRef1 (Item reference)
Local variable ItemRef2 (Item reference)
Set reference ItemRef1 to the_ref
Set reference ItemRef1 to the_ref($ref.$ident)
Calculate %%TYPE as the_ref().$objtype
If %%TYPE<>'table'
  OK message {You have to click on a table!!!}
  Quit procedure
End If
Calculate %IDENT as the_ref().$ident
Set reference ItemRef1 to
$root.$windows.[#TOP].$objs.[%IDENT]
Calculate #F as ItemRef1.$tablinestyle.$assign(16)
Set reference ItemRef2 to
ItemRef1.$horzdividers.$add(ItemRef1.$tabrheight)
Calculate #F as ItemRef2.$linestyle.$assign(1)
Calculate %%WIN as #TOP
Close window [%WIN]
Open window [%WIN]
```

*Compiled by Stephen Miller
stephen_miller@dlagroup.com.au*

Lists with Item reference variables?

In OMNIS 7.3.x, it isn't possible to store item references in lists.

For now, the workaround is to instead store each reference's \$fullname in the list, as a character field.

For example, instead of doing this:
Set current list #L1

```
Define list {varItemRef}
Add line to list {($cwind.$objs.MyObj1)}
...
Load from list {1}
Calculate varItemRef.$enabled as kFalse
```

Do this:

```
Set current list #L1
Define list {#S1}
Add line to list {($cwind.$objs.MyObj1.$fullname)}
...
Load from list (1)
Calculate [#S1].$enabled as kFalse
```

Except for the destination field in a Set reference command, OMNIS always 'dereferences' item ref vars before using them.

In other words, whenever you use an item reference var in OMNIS, it always represents the object/attribute pointed to by that var, not the var itself.

The only exception is the Set reference command, e.g.

```
Set reference varItemRef as $someNotation
```

Here varItemRef is not 'dereferenced', which is why Set reference is the only command in OMNIS that let's you change the contents of an item ref var.

*Leon Venter
Blyth R&D*

#EDATA and Modeless Enter Data

New Blyth CEO You open a window open (your "normal window") and this window calls the procedure command Enter Data. Now your OMNIS system is in enter data mode (#EDATA=1).

—While your "normal window" is open, you allow the user to click a button or do a keyboard command that opens your HELP window. However, the HELP window does not contain a procedure command "Enter data" (called from the INIT (0) procedure, or calling to another procedure in this HELP window).

If this is the case—when your HELP window opens, you have lost control of #EDATA. #EDATA is 1 (the system is in enter data mode), but your windows are not controlling enter data variables correctly. This is especially true if the open windows do not have their own WCP's that contain code that specifies an enter data action if #TOP='w_window_name'.

One way to solve this problem is to execute the procedure command "Enter data" in the Help window's procedure set. You do this if your "normal window" executes the procedure command "Enter data." When the HELP window opens, this window's INIT procedure (0) sets a WCP and executes the procedure command "Enter data." If you are using reversible blocks, then you call another procedure inside this window (e.g., Procedure/50) from the INIT procedure (0) since the procedure command "Enter data" stops program execution and does not allow reversible blocks in INIT procedure (0).

With this said, I would add that although this scenario will work (and you control the enter data variables), it is NOT what you want to do.



the omnis[®] underground communique

april 8 - 11, 1997

www.omnis-underground.com

The Omnis[®] Underground Announces 2nd Annual Technical Conference and Training Seminars

DENVER - Once again, The Omnis Underground is sponsoring "Geek Week". This annual trek by the Omnis developer community has proven to be a premier developer event. To be held at the Cherry Creek Inn in Denver, CO, the conference runs April 8-11, 1997. Steve Knouse, Director of Data Processing for Lone Star Medical Products had this to say; "The Omnis Underground Technical Conference, aka Omnis Geek Week, is the place to be. If you value your time, if you want to get the most bits for your buck, if you want to rub shoulders with some of the best Omnis people in the world, if you want to experience technical overload, Geek Week is for you. Geek Week is red meat to OmnisWorld's quiche (with no apologies to vegetarians.) Bring your beanie, your laptop and get plenty of sleep the week before. If you are serious about improving you Omnis skills this is the place to be."

The Omnis Underground, started in Denver by a group of Omnis aficionados, were forced from the mainstream and into the underground. This small band of programming renegades (actually they prefer the term zealots) meet regularly to teach and learn from each other all that they can about their passion, "Omnis". Their mission statement and charter is clear: improve the skills of the Omnis developer community.

All those who attended last November's Conference reportedly left with knowledge, tips and skills that earned them 100- to 1000-fold what it cost to attend.

What is this year's conference goals? According to our sources, the following outlines the groups goals:

- Training, Training, Training
- Create hands on technical sessions
- Assemble an incredibly rich and diverse group of session leaders
- Get in the trenches and show off serious development techniques
- Foster improved developer community relationships
- Keep attendance costs low
- Present only technical courses
- Attract all levels of developers
- Generate open, interactive sessions where the developer community can share the best and the worst Omnis has to offer

The Underground invites you to attend "Geek Week 2" and join the rebellion. Positive you will leave with new knowledge and new technical associates, The OU ask

only that you come open to share what you know, have a willingness to learn, and ask as many questions as possible.

Highlights of this years conference include:

- **Prometheus** - Full Prometheus training. Three full days of education on creating applications using this new Object Oriented development tool.
- **OMNIS 7.3.5** - Three days of Omnis 7 training. A complete course covering all of the basics. Hands on instruction from some of the best developers in the country.
- **Web RAD** - Twelve (12) training sessions covering Omnis Web RAD & Internet Tools. A variety of issues about the Internet and its new generation of possibilities. Sessions include HTML, JAVA, Web Page & Site Design and integration of Omnis with the Internet.
- **SQL** - Planned is an extensive lecture series and hands on sessions for those developers working with or breaking into the SQL arena. Four (4) sessions are scheduled for teaching the basics of SQL. Four (4) sessions of Oracle issues will be discussed in depth.

A host of world class, international instructors have been recruited to lead technical discussion groups and teach a wide variety of topics. Specialized topics include Omnis Extensions, Data File Maintenance, Deployment, Component creation, and many others.

Blyth has graciously offered attendees **special product pricing on Prometheus**. Only available at the Underground Conference, Prometheus Golden Master 1.0 will be made available to all attendees. A time limited version will be given to all. If purchased at The Underground Conference, a special discount of 15% for attendees and a 25% discount off of list price for official Underground members is offered. Purchasers at the conference will be first to receive the final release version when shipped on May 15th.

This conference appears to be shaping up to be the finest in Omnis' history. The Underground has a track record of producing quality conferences and has a firm belief in the developer community. More details of this event are at <http://www.omnis-underground.com/>

How to contact us?

The Omnis Underground
701 South. Logan St., Suite 201
Denver, CO 80209 USA
Voice 303.765.4417 Fax 303.765.4418
e-mail OU@netway.net

Official Agents Wanted

The Omnis Underground is currently drafting agents for official membership. As an agent you will receive the following benefits:

- \$50 discount on annual assignments; domestic and abroad (Conferences)
- Free official Underground Shirt or Beret
- Six bimonthly issues of "OmniDirectional"
- 15% discount on the purchase of all Blyth products. 25% discount on Prometheus at the Conference (limited to US Developers)
- 25% discounts on the purchase of all Underground products, including videos, uniforms, and accessories
- Other special offers throughout the year

All of this for just \$195.00 per year! Sign up at the conference or contact us at OU@netway.net.

Keynote Address: Where is Blyth going?

With all of the changes happening at Blyth, it is critical that the developer community be kept informed with what will dramatically affect all of our careers. Tim Negris, President and CEO for Blyth Software will present the keynote address. He will focus on the new company and the future, both short and long term. This address will prove to be an exciting glimpse into a new world.

Tim Negris joined the company in the role of Executive Vice President, Marketing and Development and since has taken the reins as President and CEO.

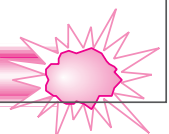
Mr. Negris joins Blyth with over 15 years of experience in software marketing. Most recently as Vice President, Sales and Marketing for IBM's multi-billion dollar Software Solutions Division, he spearheaded marketing for their award-winning database, workflow, digital media and application development products. Mr. Negris was also instrumental in establishing joint development and marketing programs with emerging Silicon Valley news media and electronic commerce ventures, as well as in affirming IBM's presence as a software provider in emerging markets such as China, Eastern Europe and Latin America.

Prior to IBM, as Vice President, Server Product Marketing at Oracle Corporation, Mr. Negris was one of the primary architects of Oracle 7's highly successful market launch. Previously he served as an executive at Sybase and a number of other software companies.

Mr. Negris joins Blyth at a strategic time, as the company readies it's new product line for entry into the hot arena of electronic commerce, while completing a major revamping of its well-established OMNIS application development tools.

Time	Track 1	Track 2	Track 3	Track 4	Track 5	Track 6
Wednesday	Prometheus	Advanced Omnis SQL	Omnis 101	Assignment Internet	Hot Diversity	In the Lab
8:30 - 10:00	Keynote - Breakfast	Keynote - Breakfast	Keynote - Breakfast	Keynote - Breakfast	Keynote - Breakfast	Keynote - Breakfast
10:15 - 11:45	Welcome to the Paradigm - OOP Concepts and Design	Beginning SQL	Defining The Game Board 1	Omnis Internet Services	Building Tools for Prometheus	Product Comparison
11:45 - 1:00 Lunch						
1:00 - 2:30	The Physical World - The new IDE	Advance SQL I	Defining The Game Board 2	Access Denied: Cracking the Web Database	Apple Script and Omnis	Product Comparison (continued)
Coffee						
3:00 - 4:30	Approaching the Architecture - Prometheus Design Considerations	Advance SQL II	Reviewing More of the Game Rules	Omnis Web RAD		Hands on Notation
Snack						
4:45 - 6:15	The Visual World	Converting from DML to SQL	A Few More Things You Need To Play The Game	Omnis Web RAD	Benchmarking with Omnis	Hands on Notation (continued)
6:30 - 7:30	Inside Blyth Tech Support					
Thursday						
7:00 - 8:45 Breakfast						
8:45 - 10:30	Communicating with Methods - Methods are our friends	Oracle - Exposing the "Giant" I	Starting the Game	Omnis Web RAD	Mysteries Myths and Gotch's Part I	Prometheus - SQL Lab
Coffee						
10:45 - 12:30	The Task at Hand	Oracle - Exposing the "Giant" II	Adding More Playing Pieces	Omnis Web RAD	Mysteries Myths and Gotch's Part II	Prometheus - SQL Lab (continued)
12:30 - 1:45 Lunch						
1:45 - 3:30	Output is Power	Oracle - Exposing the "Giant" III	Circling the Game Board the Last Time	Web RAD Possibilities Panel		Web RAD Lab
Snacks						
4:00 - 5:45	Build It, Reuse It - Creating Reuseable Componets	SQL Q & A	The Finish Line!	Now its Real (Power Computing)	Version Control System	Web RAD Lab (continued)
Friday						
7:00 - 8:45 Breakfast						
8:45 - 10:30	Let the Tool Work for You - Wizards and Other Built Ins	Data File Maintenance	Introduction to Notation	What's Cool, What Sucks	Deployment & Upgrading	Hands on Prometheus
Coffee						
10:45 - 12:30	The Outside World - External Components	Advanced Notation	Lists and an Introduction to Tables	Under the Hood Introduction to HTML	Optimizing DML	Hands on Prometheus (continued)
12:30 - 1:45 Lunch						
1:45 - 3:30	Advanced Prometheus Topics - Smart Lists, etc.	Notational Reports	Tables Tips and Tricks I	Is it Real or Memorex® VRML	Extension Development	SQL Attack: Oracle
Snacks						
4:00 - 5:45	Prometheus Q & A	Notational Q & A	Tables Tips and Tricks II	Coffee Break Introduction to Java	Extended Areas and Multi-segments	SQL Attack: Oracle (continued)

s e s s i o n s c h e d u l e





What is your time worth?

Do you have days to spend learning how to get started with Blyth's new object oriented technologies and Internet enablement tools? Wouldn't you prefer to learn from the best OMNIS talent in the world — at your convenience? The OMNIS Underground has secretly acquired these highly classified training tapes.

These tapes have been covertly filmed at the Prometheus, Web Dev and Notation sessions during the November 1996 OMNIS Underground Technical Conference, in Denver Colorado.

The tapes feature the talents of David McKay, Director of Product Marketing for Blyth Software; Bill Hamilton of WMPNJ Software; Chris Clarke, Internet Evangelist for Blyth Software; and OMNIS legend Leon Venter, of Blyth Software.

The Prometheus series features approximately 12 hours of training, covering OO concepts, basic Prometheus tools, and use of the new IDE (integrated development environment). Bill Hamilton focuses on conversion and issues related to porting 7.3.x to Prometheus. (A\$250)

Chris Clarke provides nearly 6 hours of coverage on **Blyth's new Inter/Intranet enabling tools**. These new tools will empower the OMNIS developer to access and be accessed by the World Wide Web. Web Dev enables an OMNIS application with TCP/IP protocols, opening a new world to developers. (A\$200)

Finally, we offer **6 hours with the master of Notation**, Leon Venter. Leon covers everything from the basic Notation to advanced tips and tricks to the most closely guarded secrets of a wizard of OMNIS DML. (A\$200)

Order your copies from Amanda Turner at DLA. email: <amanda_turner@dlagroup.com.au>
Tel: +612 9262-2290 Fax: +612 9262-2290.



The OMNIS Advisor

This column will be a regular feature of *OmniDirectional*, in which I will closely examine an OMNIS solution to a particular task. At the end of the column, I will pose the task to be discussed in the next column. I encourage all readers to send me their solutions, comments, and questions at jim@jpcr.com.

The Task

How can I enable my users to incorporate color into their applications?

The Solution

When thinking about color and OMNIS, I think of three aspects to the task:

- 1) Providing the user with an interface to select a color.
- 2) Saving and storing colors.
- 3) Using colors.

The Demo

To best understand this discussion, download the JPCRDEMO.LBR from the following URL:

Windows: ftp://ftp.crocker.com/pub/users/pistrang/JPCRDEMO.zip

Mac: ftp://ftp.crocker.com/pub/users/pistrang/JPCRDEMO.sea.bin

The demo is a small OMNIS library that installs a menu named Jim's Demos. From this menu you can open the Color Demo window. The demo code is unlocked, and I encourage all readers to view it, copy it, enhance it, and use it!

The Color Selection Interface

If you go to the Change Colors dialog in OMNIS Preferences, you'll see that OMNIS works with a palette of 272 colors (just to confuse matters, there are actually only 256 colors, since the top 16 are repeated elsewhere in the table). OMNIS sees each of these colors as a number from 1 to 272. The interface in my demo allows the user to click on a color in a palette and return the index number of the clicked color. It does this by knowing the precise size of the palette, and determining where the mouse is located at the time of the click. If you're feeling ambitious, step through the code and see precisely how this happens. (Thanks to the original OMNIS Graph Demo and Ron Sweet for helping to put this together).

Saving and Storing Colors

Once the user has selected a color, we've got an index number in OMNIS, ranging from 1 to 272.

There's nothing special about the number, it can be saved to and retrieved from a database just like any other number.

Using Colors

Once a color has been translated from the palette into an index value that OMNIS understands, it can be used via notation to set color attributes. In the following examples, the color index value is stored in the field fvColorCode:

To set the background color of a background object:
 calculate \$cwind.\$bobjs.1075.\$backcolor as fvColorCode

To set the color of a text object:
 Calculate \$cwind.\$bobjs.1076.\$textcolor as fvColorCode

To set the color of a button:
 Calculate \$cwind.\$bobjs.1076.\$textcolor as fvColorCode

To set the background color of a window:
 Calculate \$cwind.\$bobjs.1076.\$textcolor as fvColorCode

To set the foreground color of a background object (ident 1084) on the current row of a table (ident 1012):
 Calculate \$cwind.\$objjs.1012.\$tabrbobjjs(1084,#LN).\$forecolor as fvColorCode

Next Column

I recently made a reference in an OMNIS List posting to a 'low tech' version of the CMS that I've implemented at a client site. The OMNIS CMS is a powerful tool, and I'm not advocating that it be abandoned. However, sometimes a lesser (and less expensive) solution is appropriate. How have YOU automated the distribution of new versions? Send me your solutions at jim@jpcr.com. In my next column I'll discuss my solution, as well as those of other talented developers.

by Jim Pistrang

Jim Pistrang is the president of JP Computer Resources, based in Amberst, Massachusetts. He's a full time OMNIS developer, a Certified OMNIS Software Professional, a Blyth Ambassador, and



Proven Mission Critical Web Application Development Environment

OMNIS Web RAD

OMNISCIENT

One Integrated Tool:
pure programmer productivity.

“The coolest thing is that OMNIS lets you write code that writes code. We built an entire application generator with OMNIS that builds applications for us. No other product does what it can do.”

Scott Bergin
Denver, CO
13-year developer

Super RAD: hyper coding and debugging.

“The OMNIS debugger is absolutely the best debugger in the world. I’ve yet to find another tool with a debugger that is as sophisticated and comprehensive. It’s phenomenal for testing complex, tricky and hard-to-find bugs.”

Melissa Worthington
Minneapolis, MN
10-year developer

OMNIPOTENT

Universal Scripting Language:
any desktop, database, or Web service.

“Having one language that lets me do FTP access, CGI access, SQL access is incredibly valuable.”

Kevin Doyle
Boston, MA
15-year developer

Socket-to-Socket on the Net.

“With socket-level interfaces I have the low-level control I need to build my own Internet services, including Internet transactions.”

Dewey Gaedcke
Austin, TX
13-year developer

Direct Data Access: Informix, Oracle, Sybase—you name it.

“OMNIS is the only industrial strength development environment out there. I trust it with my critical data. And I can access any database and move data anywhere to solve a problem.”

Larry Barcot
Yorba Linda, CA
14-year developer

OMNIPRESENT

All Architectures:
client/server, peer to peer, mobile and the Net.

“The ability to connect to anything, to exchange any kind of data with anything, the ability to do anything, anywhere, at anytime. Sounds impossible, but that’s exactly why I’ve stayed with OMNIS.”

Jim Pistrang
Amherst, MA
7-year developer

All Platforms: Windows 95, NT, MacOS, and OS/2.

“When people understand the meaning of “omnis” they will grok how all-encompassing this product is. OMNIS cuts development time by at least 50%.”

Jerry Lipscomb
Denver, CO
11-year developer

OMNIS Web RAD product family allows you to create transaction-based Net applications that integrate data, messages, and external services.

Workgroup Edition

Complete OMNIS Internet development environment including the OMNIS desktop database.

Enterprise Edition

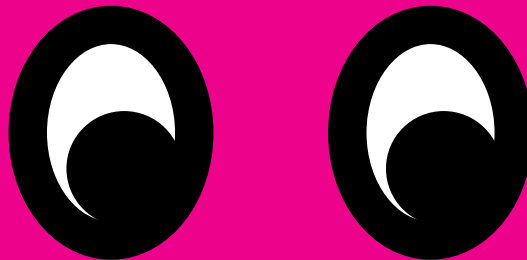
Full-featured client/server system providing the OMNIS Workgroup developer tools plus access to Informix, Oracle, Sybase and leading databases. Object management and version control, including team development support.

Go to www.blyth.com

or call 1-800-346-6647

Beyond-the-edge Tech Conference!

Marketing bozos apologize; "We knew OMNIS was good, but not this good." Cult following ensues as geek developers outsmart even themselves exploiting OMNIS Web RAD's true potential.



"The unity and networking of the Underground spark a creativity that's unparalleled in development communities. Like Deadheads, we're a true cult, with the same dedication and loyalty. We're techies who push the envelope, to do what Blyth thought couldn't be done."

"OMNIS is so powerful, so flexible, so capable that the OMNIS Underground conferences are necessary to learn all the tips and techniques. No marketing fluff. It's the down-and-dirty technical scoop, not PR spin."

The OMNIS Underground Has Surfaced

"There's lots of things I cannot do with 4D or PowerBuilder. There is nothing I can't do with OMNIS."

"Not for the faint of heart. They are pointed at real technical developers. You have some of the best talent in the world at these conferences."

HOT LINKS

Continued from page 11

I suspect that what you want to do is create a non-modal interface where you have more than one window open (many windows for that matter), and the user can jump from window to window (via clicking or executing menu commands) entering data where they want. Each open window has its own controls, and the entire system works together to produce the result that Mac and MS Windows users expect. For example, the user opens an invoice window and begins to enter lines for the invoice. Then the phone rings. It's another customer who wants to know their account balance. Your user then opens the ledger for the caller (leaving the original invoice window open—this is the window where the user is entering an invoice for the first and different customer). While this is going on, the user may need to open an inventory window and add an item, or modify the item count, or modify something about the item. You let the user do this while leaving the other windows open, all in enter data mode. If the user wants to quit, and they have not saved data in the open windows, you put up a friendly message asking if the user wants to save the Invoice, save the changes to inventory items, etc.

To create this type of structure, you do NOT use the `_procedure_` command "Enter data." Instead, you set the "Modeless enter data" bit in your non-modal data entry windows. In these windows, you control events using FCP's and a WCP. This way, each window that comes to the top resets `#EDATA` to 1 and respects its own WCP. (BTW—It's best to use a single WCP when possible since multiple WCP's make it difficult to read your code.) To test this, create a menu line in a menu that is always available (for testing purposes). The menu line contains one command:

OK Message {`#EDATA=[#EDATA]`}. Create two windows—`WIN_1` has the "Modeless enter data" bit set, and `WIN_2` does NOT have the "Modeless enter data" bit set. Open both windows, and toggle one to the top. If the top window is `WIN_1`, and you select the menu command that tells you the value of `#EDATA`, you see that `#EDATA=1` in the OK Message. If the top window is `WIN_2`, and you select the menu command that tells you the value of `#EDATA`, you see that `#EDATA=0` in the OK Message. If the INIT procedure (0) in either of these windows executes the procedure command "Enter data" or calls a procedure that executes the procedure command "Enter data," then selecting your test menu line will always say that `#EDATA=1`.

Executing the procedure command "enter data" is global, and the entire system is in `#EDATA=1` until the user selects OK or CANCEL, or closes all open windows.

The "Modeless enter data" bit in a window is specific to that window, and `#EDATA=1` only when that window is on top. Of course, if you execute the procedure command "Enter data" or you have another window on top that has its "Modeless enter data" bit set, `#EDATA` will be "1" as well.

So what the heck do you use the procedure command "Enter data" for? The procedure command "Enter data" is useful when you want to stop program execution. When you execute the procedure command "Enter data," OMNIS stops and waits for the user to enter data and terminates with an OK or CANCEL. Save this procedure command for modal windows that open and demand user input. These windows are modal (visually and functionally). A modal window

has no close box, grays the menu bar, and forces the user to do something. When you open a modal window, you execute the procedure command "Enter data" and make the user do OK or CANCEL. There may be some data entry in these modal windows (entering text, selecting radio buttons or check boxes, etc.); however, the user finishes with OK or CANCEL, and then OMNIS continues with the next command in the procedure stack. One modal window may allow opening another modal window, but the second modal window that opens also contains the procedure command "Enter data," and waits for OK or CANCEL. Again, the second modal window may contain text fields, radio buttons, or check boxes—but it concludes with OK or CANCEL.

In summary—

—The procedure command `#EDATA` is global to the application.

—Executing the procedure command "Enter data" stops program execution and waits for the user to enter data and finish with OK or CANCEL.

—If you open a window that executes the procedure command "Enter data," your OMNIS system is in enter data mode (`#EDATA=1`). Opening other windows does not affect the `#EDATA` variable (the system is still in enter data mode, that is `#EDATA=1`).

—Reserving the procedure command "Enter data" for modal windows is useful since this stops program execution and waits for user input. If you do this (execute the procedure command "Enter data"), and you open another window, you need to either add another "Enter data" procedure command to the stack or use a LCP that handles the two open windows (or three or more—all—open windows).

—If you want to create a non-modal interface where you have many data-enterable windows open at one time, set the "Modeless enter data" bit in the individual non-modal windows and use a WCP in each window to control enter data events. Reserve the global "Enter data" procedure command for non-modal windows that stop program execution.

Contributed by:

Spike Mylan

Joey Software, Inc.

Square Bracket Notation and Double Slashes (//)

Square-bracket notation can be used in several ways, e.g.:

1) In OK messages:

Calculate `#S1` as "Hello"

Ok message {`#[#S1]`} ;; Displays 'Hello'

2) For indirection:

Calculate `#S1` as "`#S2`"

Calculate `#[#S1]` as 'Hello' ;; `#S2` now contains 'Hello'

Calculate `#S3` as `#[#S1]` ;; and so does `#S3`

3) For calculated object ids in notation:

Calculate `#1` as 1001

Calculate `$cformat.$objs.[#1].$textcolor` as `kDarkBlue`

4) For calculated object names in notation:

Calculate `#S1` as 'Edit Button'

```
Calculate $cwind.$objs.[//#S1].$enabled as kFalse
OR
Calculate #1 as 1
Calculate $cwind.$objs.[//
con('RadioButton',#1)].$enabled as kFalse
```

The only case where you use a double slash is (4), i.e. when you're using square brackets in notation to generate an object name. You need the double slash to prevent OMNIS from getting confused by 'unusual' object names, e.g. those containing periods, beginning with numeric chars, etc.

This ensures that your notation will always work, regardless of the object names it has to deal with.

*Leon Venter
Blyth R&D.*

Square Bracket Notation and Calling Procedures

This works

```
Calculate #S3 as con('TEST/2 ')
Call procedure [#S3]
```

This code doesn't

```
Calculate #S3 as con('TEST/2
',' ','Param1',' ',' ','Param2',' ',' ','Param3',' ',' ','Param4',' ',' ')
Call procedure [#S3]
```

But the following code does work as the assigning of the proctext to a spare procedure slot tokenises the code which appears to be required when parameters are passed.

```
Calculate MY_NEW_PROC as con('Call Procedure TEST/2
',' ','ParamString1',' ',' ','ParamString2',' ',' ','ParamString3'
',' ',' ','ParamString4',' ',' ') (Values Passed)
```

or

```
Calculate MY_NEW_PROC as con('Call Procedure TEST/2
',' ','ParamFieldOne',' ',' ','ParamFieldTwo',' ',' ','ParamFieldThree'
',' ',' ','ParamFieldFour',' ',' ') (Fields Passed)
```

then

Procedure 450 is a procedure slot reserved for 'dynamically created proctext.

```
Calculate #F as $cformat.$procs.450.$proctext.
$assign(MY_NEW_PROC)
If flag true
    Call procedure 450
End If
Quit procedure
Local variable MY_NEW_PROC (Character 10000000)
```

*Compiled by Stephen_Miller
Stephen_Miller@dlagroup.com.au*

The Data Files are Full

The data files are full OMNIS error message occurs on MacOS. Utilities will not allow you to automatically expand the data file.

Sometimes this problem can be fixed with Utilities:

- Check Data File Structure
- Check records
- Check indexes
- Perform repairs

OMNIS expands data files automatically and after many small automatic-expands on the mac it could happens that the Finder will not expand the file again because there are too many file fragments. Utilities will not allow you to manually expand the file.

You can procedurally expand the data file by yourself (notation) or test the \$freesize of the datafile and if it is less then n KB give a warning to the user, that the df has to be expanded (manually).

This problem does not necessarily require you to export the data to text files and import into a new data file. If you have access to DOS file structure, such as on a server, a Networked PC or an external media device formatted with the DOS system, then you can use the Windows version of OMNIS to increase the size of the data file. This method is not guaranteed to work but usually does. The best approach is to expand the data file procedurally so that in the life of the data file there are simply several large increases in size instead of numerous incremental increases.

Compiled by Stephen Miller from various contributors to the OMNIS List Server.

Pasting Text

Paste text from the clipboard into an active text field at the current position of the cursor

Object name of the entry field on the window = EntryObj.
The data field associated with the entry field = EntryFld.

In proc 0 of the window, add these lines:
Format var EntryRef (Item reference)
Set reference EntryRef to \$cwind.\$objs.EntryObj

The field proc for the "Paste" button should do this:
If #CLICK

```
Local var lFirst (Long integer) = EntryRef.$firstsel
;; Save curs position
Local var lText (Char 10000000)
Paste from clipboard lText
;; Get contents of clipboard
Calculate EntryFld as con (mid(EntryFld,1,lFirst),
lText, mid(EntryFld,EntryRef.$lastsel+1,
len(EntryFld))) (Redraw field)
Calculate EntryRef.$firstsel as lFirst
;; Now select the pasted text
Calculate EntryRef.$lastsel as lFirst+len(lText)
```

End if

*Leon Venter
Blyth R&D*

