OMNIS Software announces immediate availability of the World's first crossware development tools

RAD Software Enables Complete Integration of Java Beans and ActiveX Components

FOSTER CITY, CA, May 5, 1997—OMNIS Software, Inc. today announced OMNIS Studio, OMNIS Studio Production Manager and OMNIS Studio Data Access Manager. The release version of OMNIS Studio, which has been in beta testing for nine months under the code-name Prometheus, is the first Rapid Application Development (RAD) tool to deliver crossware, software that is cross-platform, cross-database, cross-object and cross-architecture. OMNIS Studio offers a complete set of tools to integrate, optimize and manipulate best-of-breed components of all types, including ActiveX and Java Beans. Combined with OMNIS Studio, OMNIS Studio Production Manager takes version control to the next level, while OMNIS Studio Data Access Manager allows users to access and manipulate data simultaneously from all leading databases.

The initial release of these tools will be followed by additional products which extend the advantages of OMNIS Studio's component-based technology through deployment and distribution. The new products add powerful development features that help focus on their bottom line rather than cross-compatibility issues.

"OMNIS Software is introducing an environment that is highly valuable to our partners, who are developing some breakthrough applications," said Tim Negris, President and CEO of OMNIS Software, Inc. "Corporate and commercial developers need tools that allow them to use a variety of components and database types. OMNIS Studio's crossware effectively ends the component war."

The announcement of OMNIS Studio comes on the heels of OMNIS Software's recent name change from Blyth Software, Inc. "Over the last few months, we've put this company back into start-up mode, yet unlike most start-ups breaking into the market, we have real products available today and real people using them," said Tim Negris, who was appointed President and CEO of OMNIS Software, Inc. this past quarter. "This company has great assets, both in the OMNIS technology and in the wide base of customers and VARs who have deployed OMNIS applications to almost one million end users."

Negris said the change in identity forms the basis for an aggressive new program promoting a breakthrough new generation of application development products built on the OMNIS technology and extending new capabilities to the existing customer base.

Hundreds of OMNIS partners are implementing solutions around the world. Tens of thousands of corporate and commercial developers use the OMNIS environment. Because of this unique cooperation with partners and VARs, tens of thousands of applications have been deployed using OMNIS technology. Over one million end-users have OMNIS-based applications deployed on their desktop.

OMNIS Studio Provides Complete Built-In Component and Web Integration

OMNIS Studio enables the importation, assimilation and extension of objects including Java, ActiveX, OCX, VBX, C++, OMNIS and other native component types. OMNIS Studio interrogates the properties, methods and events of each object, and represents them to the developer in such a way that they may be extended or executed. It also enables the application to

communicate via a full set of Internet protocols including Mail (SMTP/POP3), File (FTP), Web Server (HTTP) and low-level TCP/IP sockets programming access.

Universal Scripting Language

OMNIS Studio uses OMNIS Script, OMNIS Software's own Universal Scripting Language, which allows developers to manipulate components and objects of any size or complexity quickly and easily. OMNIS Script combines two powerful and complementary languages:

A hierarchical, fourth-generation language designed for manipulating course-grained objects and components.

Dot Notation, a self-articulating language that describes itself or any other object or class down to individual events, methods and properties. It also allows programmers to establish meanings between components.

Features of OMNIS Studio

OMNIS Studio is a complete component design and Web integration system, providing customers and partners with the industry's first crossware Rapid Application Design (RAD) tool. OMNIS Studio can be deployed on a wide variety of platforms including Windows 95, Windows NT, Windows 3.1, MacOS and OS/2 with future support for HP/UX and Sun Solaris. Easy conversion protects investment in other OMNIS products. Its powerful notation inspector provides design and run-time interrogation of all objects, regardless of source or type. Wizards and built-in objects offer unprecedented ease in creating report applications. Other features of OMNIS Studio include:

Hundreds of Intelligent Classes and Objects - Data access, multi-window implementation, transaction processing and Web development.

Hybrid Compiled/Interpreted Code Execution - Allows developers to toggle between design and runtime modes, constantly testing their work.

Local and Portable Data Access - OMNIS Studio Cache, a relational database provides four gigabytes of local data caching in a tiny 200K footprint.

Multiple Debugging Modes - Developers have access to a comprehensive range of sophisticated debugging operations.

Support for Localization and Multi-Lingual Implementation - Provides a single environment for asset management and updating.

Features of OMNIS Studio Production Manager

OMNIS Studio Production Manager works with OMNIS Studio and serves as a central repository for objects, allowing them to be stored in any OMNIS application. Developers can work on applications from any platform, in the GUI native to that platform. Its version and tracking control allows developers to quickly roll back individual changes, retreat to previous tags or split development efforts into separate tracks. It provides detailed reports of all coding transactions, enabling team leaders to supervise individual progress and manage workflow across the team. OMNIS Studio Production Manager also provides security by allowing team managers to limit access to individual objects or entire applications, customizing privileges as necessary.

Features of OMNIS Studio Data Access Manager

OMNIS Studio Data Access Manager works with OMNIS Studio to offer a single powerful tool to manage and access databases. Users interact with all leading databases including Oracle, Sybase, Informix and Microsoft SQL Server, using a standard set of functions and tools. This allows them to move items from one database to another with drag-and-drop ease. Developers can customize all aspects of any database including tables, views, triggers, rules, stored procedures, synonyms, storage allocations and all other classes and methods supported by back-end databases. Using universal SQL Scripting, OMNIS Studio Data Access Manager allows developers to combine data sources in a single application with ease. OMNIS Studio Data Access Manager also provides dynamic connectivity to most data sources and hundreds of other relational and legacy database systems.

Pricing and Availability

All products will be generally available on May 12, 1997.

OMNIS Studio - The Future of OMNIS!

On May 5 OMNIS Software, Inc., formerly Blyth Software, announced the release of the new versions of OMNIS under the banner: OMNIS Studio. Several new initiatives accompany this release with the most significant being the integration of Web tools with the OMNIS suite of products.

Below we set out a detailed analysis of the new products, their features and benefits.

The DLA Group announces immediate availability of these products.

The DLA Group
Sydney, Australia
info@dlagroup.com.au

OMNIS Studio

Product Descriptor: The complete component engineering and integration system

OMNIS Studio: Description

Component technologies are radically changing the application development process, yielding dramatic reductions in the cost of developing mission-strategic applications — and thereby transforming the economics of conducting business in the electronic age.

Hundreds of thousands of Java Beans, Active X, and OMNIS components are being assembled in software factories around the world. Each one of these precision-engineered components represents a substantial investment — and tremendous business potential. How can this potential be exploited? What assembly line can handle this vast and expanding inventory of re-usable parts?

Until now, components created in different object languages, most notably Java and Active X, have existed as isolated universes. Each of these object models promise productivity and profitability, but at a high price. Just as they once faced the fork in the road between Mac and Windows, businesses are now being forced to choose sides and submit to the tyranny of a single component type.

For programmers, these isolated universes mean countless hours of drudgery and frustration trying to make incompatible components work together.

For software managers, working with third-party components spells lengthy and high cost development cycles, and a heightened risk of project failure.

For executives, the component war means keeping one eye trained on a shifting target — today's hot product that might be tomorrow's software dead-end — instead of focusing on real business objectives.

OMNIS Studio heralds the end of the component wars.

For the first time, businesses can fully exploit the potential of all component types in a single application. OMNIS Studio provides the one environment designed to accept all components, allowing you to integrate, extend, and deploy them at will.

With OMNIS Studio, you can build anything, anywhere.

OMNIS Studio: Key Product Messages
I. OMNIS Studio is the first and only RAD tool to offer cross-object compatibility and customisation

Whether it's Java Beans, Active X, or a native component, OMNIS Studio offers a level playing field and a first-rate set of tools to integrate and optimise components of all types, for deployment anywhere on any platform (see Figure 1 below).

No other tool gives you the power to do so much with all component types. Instead of choosing one object paradigm over another, developers can have all best-of-breed components in one application. And unlike other applications that *claim* to bridge the component gap, OMNIS Studio was expressly designed as an open and extensible framework for building, customising, and integrating components quickly and easily.

Figure 1: OMNIS Studio provides a workspace and tools to integrate all component and data types into a single application. OMNIS Studio-built applications can be developed, deployed, and distributed across all architectures on all platforms.

What's the secret? The genius here is OMNIS Script — our own Universal Scripting Language — a powerful software invention that realises the dream of object standards like CORBA, COM, and DCOM.

OMNIS Script actually masks the differences between components so that developers can focus on achieving their own development objectives. Instead of learning the syntax of events, properties, and methods for each component, OMNIS Script allows you to use one standard language to control any and all components. This gives you the breathing room to concentrate on *what* you want a component to do, rather than *how* to do it.

II. OMNIS Studio is a world-class object-oriented rapid application development tool

OMNIS Studio is an unparalleled RAD tool, offering a vast array of features to facilitate rapid application development in a practical, visual design environment. With OMNIS Studio, developers enjoy the best of both worlds: the ease and re-use potential of object-oriented programming, along with the ability to go under the hood and edit or write code.

Local and Portable Data Access
OMNIS Studio ships with a personal copy of OMNIS Studio Cache, a relational database providing 4 gigabytes of local data caching in a tiny 200K footprint.

Easy conversion from other OMNIS products
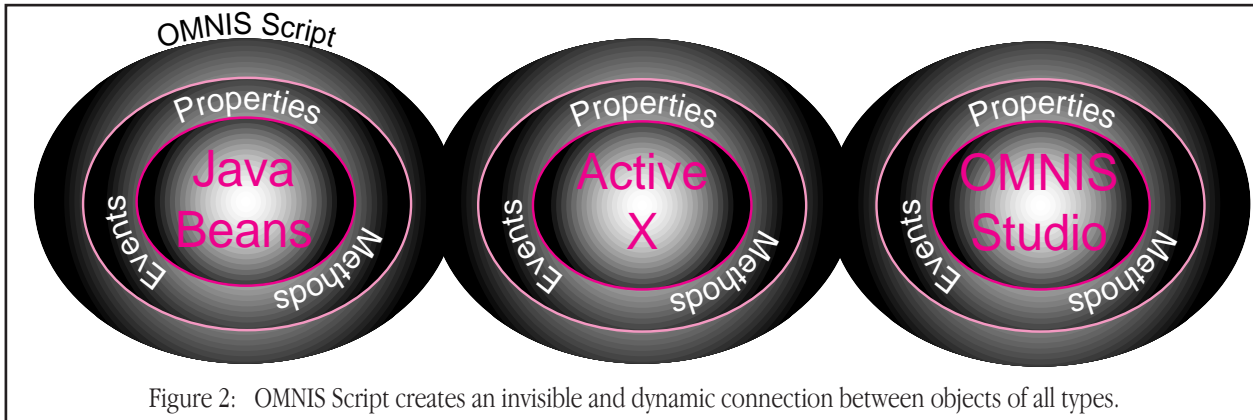Protects investments in existing OMNIS applications.

Figure 2: OMNIS Script creates an invisible and dynamic connection between objects of all types.

OMNIS Studio's powerful development toolkit includes:

Cross-platform client support for Windows 95, Windows NT, Windows 3.1, MacOS, OS/2, and coming soon: HP/UX and Solaris.
Eliminates cross-platform incompatibility as a barrier to developing, deploying and distributing applications.

Hundreds of intelligent classes and objects
OMNIS Studio offers countless possibilities for the object-oriented programmer, covering a vast array of application needs including data access, multi-window implementation, transaction processing, and web development.

Universal Scripting Language
OMNIS Script is a Universal Scripting Language that combines two powerful and complementary languages: a hierarchical, fourth-generation language designed for manipulating course-grained objects and components ... and dot notation, a self-articulating language that describes itself or any other object or class down to individual events, methods, and properties.

OMNIS Script allows developers to manipulate components and objects of any size or complexity quickly and easily in OMNIS Studio, handling many repetitious programming tasks behind-the-scenes. This conserves development time and allows programmers to focus on the unique challenges in each application they build.

Hybrid Compiled/Interpreted Code Execution
Allows developers to toggle between design and runtime modes, constantly testing their work. Optimises performance of both OMNIS Studio objects and third-party components, shortcutting development cycles by eliminating time spent compiling and recompiling applications.

Powerful Notation Inspector
Provides design and run-time interrogation of all objects, regardless of source or type.

Unparalleled Report Writer
Wizards and built-in report objects make short work of report writing.

Multiple Debugging Modes
Developers have access to a comprehensive range of sophisticated debugging operations.

Support for localisation and multi-lingual implementation
Provides a single environment for asset management and updating.

III. OMNIS Studio provides complete built-in web integration
Web development has become a key issue in creating valuable business applications. Despite this fact, most development environments still require the purchase of additional products to develop and deploy web-enabled applications.

OMNIS Studio is alone in offering both Web Integrator and Component Integrator subsystems as standard features with every development seat.

OMNIS Studio provides complete built-in support for multiple web protocols, including SMTP/POP3, FTP, HTTP, and TCP/IP, enabling web developers to take full advantage of the web's business potential.

IV. Supported by a powerful suite of plug-ins, OMNIS Studio provides an integrated solution for software development, deployment, and distribution.

OMNIS Studio is much more than a simple RAD tool. It is a multi-faceted product suite offering solutions for every phase of the software Solution Supply Chain.
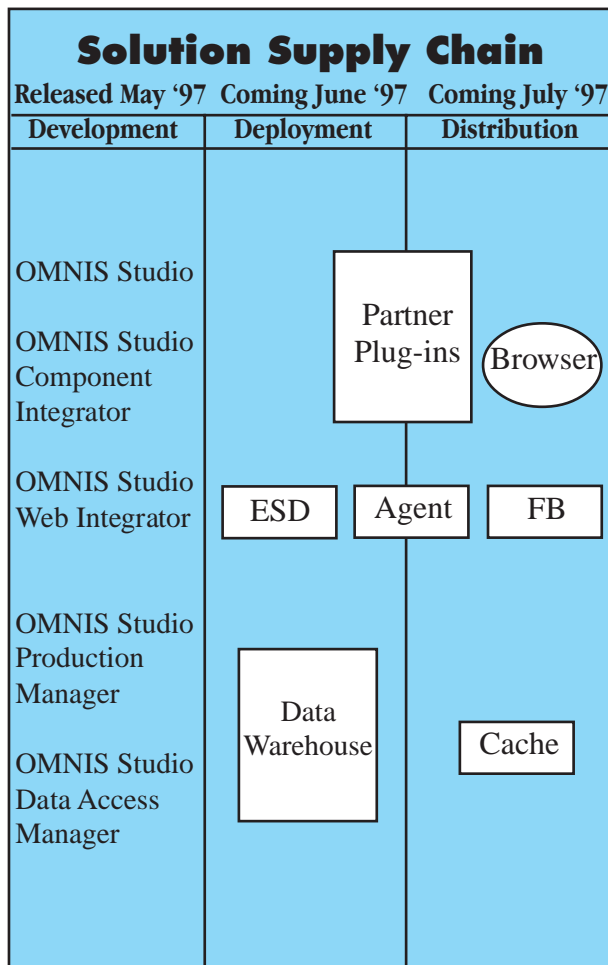
Solution Supply Chain

Released May '97	Coming June '97	Coming July '97
Development	Deployment	Distribution

OMNIS Studio

OMNIS Studio Component Integrator

OMNIS Studio Web Integrator

OMNIS Studio Production Manager

OMNIS Studio Data Access Manager

Partner Plug-ins

Browser

ESD Agent FB

Data Warehouse

Cache

Figure 3:   OMNIS Studio and its suite of plug-ins present a Solution Supply Chain.

At the core of the suite is OMNIS Studio itself, with its built-in Component and Web Integrators (see Figure 3).   OMNIS Studio Production Manager and OMNIS Studio Data Access Manager round out the development capabilities of OMNIS Studio, offering universal data access and complete control over the software process.

The initial release of these three development tools will be followed by additional plug-ins, extending the advantages of OMNIS Studio's component-based technology through deployment and distribution.

These plug-ins are the products of an unprecedented virtual software factory — a collaborative effort between OMNIS Software and its business partners to exploit the full potential of component engineering.

Together with its partners, OMNIS Software offers a single, well-integrated solution to controlling and directing the entire software supply chain.

OMNIS Studio: Positioning and Differentiators
For Programmers: BETTER!!!!!!!!!!!!!!!
1. Enables rapid object design and universal component reuse.
2. Allows programmers to focus on unique challenges of each application, eliminating many time-consuming and repetitious tasks.
3. Provides access to underlying code via a Universal Scripting Language.

4. Hybrid compiled and interpreted code technology allows for toggling between design and runtime modes without time-consuming compiles.
5. Built-in debugger.

For IS Managers:
1. Cuts costs.
2. Speeds delivery and updating.
3. Reduces risk.
4. Leverages investments in existing components and data types.

For Businesses:
1. Makes the hard stuff easy.
2. Keeps the focus on real business goals.
3. Offers unique opportunities for businesses to connect with and retain customers/end users.

For Analysts:
1. A technical breakthrough in software engineering.
2. Missing link in realising software component reuse.
3. Transforms the economics of application development.

For the Press:
1. Ends the object wars between Microsoft and Sun.
2. Transforms the economics of application development.
3. Proves that OMNIS Software, Inc. is an innovator and growth company

---

OMNIS Studio: Pricing and packaging
OMNIS Studio   A$2,299 (Quantity 1)

OMNIS Studio Data Access Manager

Product Descriptor: The universal dynamic-access database plug-in

OMNIS Studio Data Access Manager
OMNIS Studio Data Access Manager is a powerful plug-in to OMNIS Studio, providing dynamic access to all database types. Using the DAM plug-in with OMNIS Studio enables developers to access all data types — all in one integrated design environment — and know that their applications can survive a changing landscape of data sources.

Universal SQL Scripting
The secret behind universal data access lies in OMNIS Script, which handles all transactions with back end databases, masking the complexity of the exchange. OMNIS Script allows developers to combine data sources in a single application with ease. Instead of manually coding connections to each database type individually, developers can create applications and let OMNIS Studio do the rest.

Direct Access to Leading Databases via High Performance Drivers

OMNIS Studio Data Access Manager provides direct connectivity to Oracle, Sybase, Informix and MS SQL Server databases, ensuring native-quality high performance in all these leading data sources.

Common Middleware Access to all other Data Sources

ODBC provides access to most data sources using drivers from Visegenics, Intersolv, and many others. Information Builder's SQL allows access to hundreds of other relational and legacy database systems.

A Single Powerful Tool to View and Manage all Databases

The SQL Browser provides visual access to all databases, regardless of type. Users interact with all databases using a standard set of functions and tools. OMNIS Studio DAM also provides full DBA functionality unequalled in off-the-shelf tools. With OMNIS Studio DAM you can manage and customise all aspects of any database, including tables, views, triggers, rules, stored procedures, synonyms, and storage allocations. Transparent cross-data type access allows you to move items from one database to another with drag-and-drop ease.

OMNIS Studio DAM Unlocks Features

With OMNIS Studio DAM installed, developers unlock OMNIS Studio schema and table classes for defining and managing data from any backend server. Developers can choose from a wide range of standard methods for select, fetch, insert, update and delete, along with bulk data methods to handle arrays of information, or override these defaults at will.

OMNIS Studio Data Access Manager: Positioning and Differentiators

For Programmers:
1. Allows programmers to focus on unique challenges of each application, eliminating many time-consuming and repetitious tasks.
2. Provides access to underlying code.
3. Provides a single Universal Scripting Language.
4. Provides a single tool for managing and presenting all data.

For IS Managers:
1. Cuts costs.
2. Speeds delivery and updating.
3. Reduces risk.
4. Leverages investments in existing components and data types.

> **OMNIS Studio: Pricing and packaging**
> OMNIS Studio Data Access Manager   A$1,899 (per user for all databases)

OMNIS Studio Production Manager

Product Descriptor: The ultimate in software process control

OMNIS Studio Production Manager

OMNIS Studio Production Manager takes version control to the next level, providing information and tools to make production and deployment more efficient.

Power programmers, development teams, and software managers alike can now enjoy complete control over their projects, teams, and product releases with OMNIS Studio Production Manager. This powerful plug-in to OMNIS Studio provides version tracking and control, build management, and librarian services.

Cross-platform collaborative development

With OMNIS Studio Production Manager, transparently cross-platform collaborative development is finally possible. Developers can work on applications from any platform, in the GUI native to that platform. OMNIS compensates automatically for differences between platforms, so that developers can focus on achieving business objectives instead of chasing cross-platform incompatibilities.

Version Tracking and Control

Using OMNIS Studio Production Manager, team members can recall any revision of a development object at any time. Detailed version histories document changes object-by-object and build-by-build. This modular arrangement allows developers to quickly roll-back individual changes, retreat to previous tags or split development efforts into separate tracks.

Robust Production Management Capabilities

Production Manager allows OMNIS Studio users to check out code object-by-object, enabling them to work independently with confidence. Acting as a comprehensive library service, Production Manager also provides detailed reports of all coding transactions, enabling team leaders to supervise individual progress and manage workflow across the team.

Component Storage and Security

OMNIS Studio Production Manager serves as a central repository for all objects, no matter how far-flung your development effort. Objects stored here can be accessed and reused in any OMNIS application, saving time and maintenance effort across multiple projects. Team managers can also limit access to individual objects or entire applications using Production Manager, customising privileges as necessary.

Scale Up With Confidence

No matter what the size of your development effort — whether you are a lone programmer working on a single application at multiple workstations or a global team of developers working on multiple projects — OMNIS Studio's Production Manager provides a single hub for managing the development process, allowing efficient re-use of objects and well-oiled coordination of software production.

Taking the guesswork out of application builds

OMNIS Studio Production Manager organises and automates the entire build process, maintaining a complete history of build tags and object revisions all in one integrated environment. Automatic labelling of all files and tags take the guesswork and drudgery out of building applications, making this a must-have tool for buildmasters.

OMNIS Studio Production Manager: Positioning and Differentiators

1. A must-have tool for buildmasters: the only integrated tool for managing every aspect of the build and release process.
2. A single tool to manage every aspect of the application life cycle
3. Support for all object types across all architectures, including the web
4. Scalable to projects of any size and complexity
5. A must-have tool for power programmers working on multiple projects

Pricing and packaging

OMNIS Studio Production Manager   A$1,099 (per user - Quantity 1)

Creating a Demonstration Datafile from a Production Datafile

You have to explicitly set the current DF and each DF has its own CRB. The following code will do the trick.

```
Set current data file {NEW} Set read/write files
{AccountCode} Set current data file {OLD} Set main file
{AccountCode} Find first While flag true
        Set current list lvOldDataList
        Define list (Store long data) {AccountCode}
        Add line to list
        Set current data file {NEW}
        Set main file {AccountCode}
        Load from list {1}
        Prepare for insert with current values
        Update files
        Set current data file {OLD}
        Set main file {AccountCode}
        Next
End While Quit procedure Local variable lvOldDataList
(List)
```

Contributed by:
Clifford Ilkay dinamis@interlog.com

Why are my OMNIS Fonts not available on Windows 3.1 when they have been properly installed?

If you are using Windows 3.1 the system has been set to show only TrueType fonts (setting is carefully hidden in the Font Manager under options.

Contributed by: Mike Rowan
michael.rowan@adelaide.on.net

OMNIS Software names industry veteran Peter C. Mork as Vice-President of Global Sales

Foster City, CA, April 28, 1997- OMNIS Software, Inc. today announced the appointment of industry veteran Peter C. Mork as vice president of global sales. Mork will oversee OMNIS Software's product and services revenue throughout North America, Europe and Asia-Pacific. With over 20 years of sales experience, Mork joins OMNIS at a time when the company is aggressively expanding its international presence through increased global market development activities.

"Peter brings proven sales management and industry expertise to OMNIS during this critical and exciting rebirth of the company," said Tim Negris, president and CEO of OMNIS Software, Inc. "Years ago, when I was at Sybase, I had the privilege of knowing Peter. He's the guy who made Sybase a leader in the client/server enterprise arena with the customers he brought in. He has extensive experience managing sales operations across all of the product categories that OMNIS competes in and will be a tremendous asset in increasing our visibility and effectiveness in the market. He's a leader and we're looking forward to having him on our management team." "I'm excited to join a company with such great technology and extremely loyal customer base," Mork said. "OMNIS has an enduring reputation among developers and VARs that has led to applications being deployed to over one million desktops. With the exciting next-generation series of products that OMNIS is bringing to market, I believe there is a great opportunity for this company to dominate a new age in application development."

Well known throughout the software industry, Mork began his executive career with Sybase, Inc., as vice president of sales, North America, from 1986 to 1990. It was there that he developed Sybase as a major player in the database industry and was responsible for expanding revenues from $0.3 million to $70 million. Mork was among the first in the software industry to recognise the importance of establishing third-party consulting and systems integration. His success at establishing strong relationships with customers in the vertical markets, allowed Sybase to gain prominence in the financial, government, telecommunications and manufacturing industries.

Succeeding Sybase, Mork was president and CEO of Aurum Software, a leading provider of customer relationship management software. During his tenure at Aurum, Mork established "Customer Resource Planning (CRP)" positioning for enterprise-focused client/server packages addressing telesales, telemarketing, field sales, help desk and customer service. Additionally, he grew revenues from $1 million in 1991 to $5 million in 1993.

In 1995, Mork joined Persistence Software, a Silicon Valley start-up company with object-oriented middleware products, as vice president of worldwide sales.

OMNIS.

the OMNIS underground communique

www.OMNIS-underground.com

Geek Week 2 becomes a World Wide Developers Fest

We are now fully recovered from another successful "Geek Week". Rave reviews of the second annual OMNIS fest are abundant. Held in Denver, CO, April 8-11, this annual gathering of some of the best OMNIS talent proved once again that OMNIS is a world class development environment.

Tim Negris, President and CEO of OMNIS Software, Inc. (formerly Blyth Software) launched this years' event in dramatic style with a rousing keynote. This session featured a special production of the "Blyth" story staring that genius, David Seaman, the creator of OMNIS. Tim highlighted his future plans and direction for the company. OMNIS Software has an aggressive product release schedule and many new technologies are being released. The excitement level from OMNIS Software will be staggering.

OMNIS Studio (Prometheus), OMNIS Software's new flagship product was the hit of the conference. A full three days of training occupied an entire conference track. Studio speakers included Brian O'Sullivan, Mark Phillips, David Calvert, Mic Cross and Ray Barry: all of OMNIS Software, Inc. Several special guests from Mitford House in the UK (the primary R & D facility for OMNIS) came and lent their expertise to the festivities. These sessions were by far the most heavily attended. Developers from around the world came to get a jump on the learning curve of OMNIS' future.

In addition to the OMNIS Studio track, we featured a track dedicated to teaching the developer community about Inter/Intranet issues. These 12 sessions presented many issues of integrating OMNIS and the World Wide Web. We were blessed with OMNIS Software's Internet evangelist, Chris Clarke in his last appearance as a "Blyth" employee. Chris has decided to continue his career with a new JAVA based internet startup company in the Bay Area. Chris lead the charge with four sessions dedicated to training OMNIS Web RAD, the new Internet enablement kit for OMNIS. Ziff Davis Communications sent us two special guest speakers. John Hargrave lead an excellent session on integrating large databases on the Web and Genevieve Martineau gave us glimpse into Virtual Reality Markup Language (VRML). Sarah Kornfeld, of OMNIS Software, taught the "How to's" of creating a fully rounded Web site. Brett Downen helped would be web page designers begin coding in HTML. Other Internet topics included an open discussion of JAVA by Ken Crismon and Sebastian Ernest discussed how he has been integrating OMNIS into the web.

In the tradition of helping those beginning OMNIS developers to be come better programmers, faster, an entire track was dedicated OMNIS 101. With the unprecedented talents of David Swain, Leon Venter, Steve Paschke, and Scott Bergin leading the charge everyone came away with a new found knowledge base. Comments from those attending ranged from "Wow!!!" to "I learned so much my brain hurts".

Not to leave out the would be client/server developers and those who seek the "leading edge", we created a track dedicated to SQL and advance OMNIS topics. As one would expect, Jim Pistrang taught three unbelievable SQL basics sessions and Warren Anderson brought down the house with the basics of Oracle and OMNIS. Leon Venter, dazzled us with notation beyond our $wildest$imagination and Fred Haislmaier made moving from DML to SQL look easy.

Additional sessions included DLA's own Stephen Miller on Deployment and Upgrading as well as Data file maintenance; two sessions I wish I had not missed. David Ferri explored the "ins" and "outs" of the VCS and of course, no exploration of OMNIS could be complete without Kelly Burgess teaching

externals. Dewey Gaedcke, Clifford Ilkay, and Craig Lewis all added to the overwhelming brain drain of the conference.

Tracks 1-4 of this conference were video taped. We are making these tapes available to the Omnis community. See the pricing section below for details. Contact The DLA Group in Australia <info@dlagroup.com.au> or the Omnis Underground to place an order.

In other news the Omnis Underground and it's membership proudly sponsors the Omnis list server. This free daily communications forum for the Omnis developer is open to all who wish to subscribe. Currently boosting a subscribership of over 400 developers from around the world, this service provides some of the best technical tips, advice and open discussion of any community on the Internet. To join this service simply send a message to:

```
LISTSERV@omnis-underground.com
```
The body of the message should read:
```
subscribe OU_DROP_POINT
```
You will then receive a mail welcoming you to the list and providing additional instructions on how to use the list. Digest services as well as monthly archives are fully supported. We openly invite you to join and share your knowledge and questions with the community.

The Omnis Underground's Web site is designed to be a resource for you and the community. The site includes information about the Underground and it's activities. It has full FTP services and it's archives include many of the lecture notes from the conferences, sample libraries, a code repository and much more. This site is designed to help you and the Omnis development community share information and tools. Check it often. If you would like to contribute to the site or have suggestions please let us know. The URL is <www.omnis-underground.com> with the hyphen.

Underground Technical Conference Video Tapes

Video Tape Packages by Track	Tape Numbers	Package Price US$	Underground Member Price
Prometheus Track - 12 tapes	PR-1 to PR-12	$319.00	**$239.25**
Internet and Web Dev Track - 10 tapes	AI-1 to AI-10	$269.00	**$201.75**
Omnis 101 Track -12 tapes	OM-1 to OM-12	$319.00	**$239.25**
Advanced Omnis - 14 Tapes	AO-1 to AO-14	$369.00	**$276.75**
Total Price for All 48 Tapes: The above 4 Tracks =		$1276.00	$957.00
Whats your time worth? Buy all tapes and save a bundle!!!!!			
Buy all 4 Tracks & get Package 7 for the bounus price of:		**$1195.00**	**$995.00**
Over 90 hours of instruction! 56 tapes total!			
Special Video Tape Packages			
1. SQL Track - 8 tapes	AO-1 to AO-8	$239.00	**$179.25**
2. Notation & Mysteries-6 tapes	AO-10 to AO-14 & OM-9	$179.00	**$134.25**
3. Begginning thru Advanced SQL - 3 Tapes	AO-1 to AO-3	$99.00	**$74.25**
4. Oracle Exposing the Gaint - 3 Tapes	AO-5 to AO-7	$99.00	**$74.25**
5. Myths and Mysteries - 2 Tapes	AO-13 to AO-14	$69.00	**$51.75**
6. All Notation Tapes - 4 Tapes	AO-10 to AO-12 & OM-9	$129.00	**$96.75**
7. Promethues Track from 1st Conference - 8 Tapes	PROM-1 to PROM-8	$169.00	**$126.75**
Buy any 2 of the Special Video Tape Packages (#1-7) and get the third for 1/2* price!!!		**All prices are in US$**	

Shipping and Handling is **NOT** included. Individual session tapes are available for $39.95 per tape

Copyright, Piracy & Business Ethics

In this excellent article, Jim FitzSimons, a partner in the international law firm Clayton Utz, explains the world of copyright with particular relevance to computer programmers.

During the last few months DLA has pressed claims against several OMNIS programmers and other clients who have persisted with breaches of copyright. In some instances legal proceedings were commenced and in all cases these matters were settled and the client either paid for the software or removed unlicensed copies from relevant workstations.

It is clear that many companies and individuals remain unaware that software is vigorously protected by the laws of copyright and that strict obligations apply.

Jim has kindly offered to receive emails from OMNIS developers who require more information. Please also do not hesitate to contact DLA to ensure that your licences are up to date and valid for all your deployments.

What is subject to copyright?

Copyright law can protect any writing, computer program, drawing, photograph, painting (as other artistic expression) or music, however it may be stored (in a computer medium or hard copy), and regardless of whether anyone may think the "work" (as it is called) is clever or artistic. It also extends to films, recordings, radio and TV broadcasts, and even the typeset arrangements of printed material.

It does not matter whether the material is domestic or professional, for business or pleasure. An internal memo, a newspaper column, home snaps and videos are just as much subject to copyright as are computer programs, works of art hanging in galleries or cinema classics.

Also, because of international treaties which Australia has with almost every country in the world, material created overseas is just as much subject to copyright in Australia as material created by local authors.

There is no need to register copyright or to display any copyright notice, such as "©", for copyright to subsist, although it is sometimes a good idea to use a copyright notice to put others on notice of your rights.

What is copyright?

Copyright is a bundle of rights including the right to reproduce a work, whether by photocopying, retyping or recording in a different medium, and the rights to adapt or translate a work.

Copyright protects the material form in which the work is expressed, but it does not protect the underlying ideas or concepts, so that the same idea can be expressed differently without infringing copyright.

Copyright lasts for a long time, at least 50 years in photographs, films and sound recordings, but much longer in writings, artistic works and music (it lasts until 50 years after the death of the author, or the last surviving author in the case of joint works).

Who owns copyright?

The copyright owner is usually the author, that is to say the person who wrote the work rather than a person who gave an interview, made comments, or edited it. If the author produced the work in the course of his or her employment, the copyright owner will be the author's employer. The exception is journalists, who retain ownership of copyright in their work, although, their employers are entitled to publish their work once.

Consultants and other independent contractors are not employees, and so they retain ownership of copyright in their works unless they specifically agree to assign it to the person who commissioned it or engaged them. Copyright can be assigned quite easily: the only requirements are that the assignment must be in writing and signed by the person assigning it.

How is copyright infringed?

Copying of a reasonable portion of work is permitted by the "fair dealing" provisions of the Copyright Act for the purposes of research, study, criticism, review, reporting news or providing professional legal advice. This does not include copying for the purpose of supply to clients, or even for internal records or reference.

Otherwise, a person will infringe copyright when they do any of the acts within the exclusive rights of the owner without the owner's consent, whether the copying is direct or indirect, deliberate or unintentional. A copyright notice, such as "©", puts the reader on notice that a person claims copyright in a particular work, but their work can also be infringed if there is no "©" attached.

The Copyright Act also prohibits false attribution of authorship or the use of another person's name in a way which implies that he or she is the author, when in fact he or she is not. These are part of the so-called "moral rights" which is an area expected to develop greatly in copyright law over coming years.

Obtaining the copyright owner's consent

Mere ownership of a book, issue of a journal or other work does not imply that the copyright owner consents to copying. The owner's consent can be obtained quite informally, but it is best to obtain it in

Jim Fitzimmons

writing. The consent should state that the owner consents to the reproduction of the work for any purpose, or, if that is not possible, for a specified purpose. It would be best for you if the owner does not restrict the number of times that the work may be reproduced.

Obviously, it can be very time-consuming and expensive to obtain the consent of some authors to reproduction of their work. Sometimes it will not be possible to locate the author. A general licence from a copyright organisation acting on behalf of many authors may therefore be attractive.

What are the consequences of infringement?

Copyright owners whose copyright has been infringed can claim an injunction, restraining publication, and/or damages for lost revenue. An injunction, or threat of an injunction, could seriously disrupt your marketing efforts if the relevant work is part of a larger publication.

Claims for damages and accounts of profits are also available, but may be very small, because the market rates for many articles and other items are quite low. However, breach of copyright is one of the few areas in our law in which a remedy of punitive damages, is available, if the infringement is a cynical and deliberate attempt "to get something for nothing".

It is a criminal offence, among other things, for a person to distribute articles for the purpose of trade where he or she knows, or ought reasonably to know, that the article is an infringing copy. If a work includes a copyright notice, such as "©", the person ought reasonably to know that copyright subsists in it and that the copyright owner's consent is required to copy it. Fines may be imposed for this offence up to $2,500 per infringing copy for the first offence by a company ($500 for individuals), and where claims are brought in the Federal Court, a fine of up to $250,000 ($50,000 for individuals) may be imposed.

Acts which constitute infringement of copyright

All of the following acts constitute a breach of the Copyright Act, some are very much in line with the expectations of most people; others will be a surprise to some. The courts, however, do not recognise such a distinction and the penalties specified above apply equally to all breaches.

(i)   Copying a computer program and giving the copy to a friend or colleague for use by them.

(ii)   Placing a copy of a program on a hard disc accessible by users of a network when only a single copy has been licensed.

(iii)   Importing a copy of a computer program into Australia with the intention of reselling the copy. This is an infringement whether the copy is made illegally in a country which has slack enforcement of intellectual property laws or whether the copy has been bought at full price from a legitimate dealer in the United States.

(iv)   Placing a copy of a program on a bulletin board without the permission of the copyright owner.

(v)   Distributing software which you genuinely believe is in the "public domain" if in fact the author has never said that's what he or she intends.

(vi)   Making a copy of a program you use at work for use at home, unless your licence agreement permits you to do so.

Conclusion

Copyright law is not always intuitive but in the main there are some straightforward rules which can be followed to avoid any problems. Probably 90% of the advice on copyright which I supply to clients is based on the information set out in this article.

The current copyright regime may not be perfect but it is likely to remain the primary method by which intellectual property rights in software are protected and it is important therefore that everybody in the industry understands its strengths and limitations. Please feel free to email any questions to me.

James FitzSimons
jfitzsimons@claytonutz.com.au
Clayton Utz
Sydney, Melbourne, Brisbane, Perth, Canberra, Darwin

Submissions to OMNIDirectional

If you have an article, case study, letter to the editor or hot tip, why not submit it to OMNI*Directional* 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!

Our deadline for articles or advertising features for the next few issues are:

May/June Issue	June 18.
July/August	August 18.
September/October	October 18.
November/December	December 18.

For more information, contact info@dlagroup.com.au

Tips and Tricks

To restart, or not to restart…
The trick to restarting Windows 95 without rebooting your entire system is to simply choose Shut Down in the Start menu, select Restart the computer?, then hold down Shift as you click Yes.

Restarting the easy way
In the previous tip we showed you how to restart without really restarting... DO YOU WANT MORE? How about an icon that does all this without ever setting foot in the Shut Down dialog box? Just set up a simple MS-DOS batch file and place its icon within arm's reach. From then on, restarting Windows 95 is just a double-click away. Open Notepad and type the following:

```
@EXIT
```

Save the file anywhere you want with an appropriate name, such as RESTART.BAT, and close Notepad. Find the new file in Explorer and place a shortcut to it on the desktop (assuming you want to access this icon from the desktop). Close Explorer and rename the new shortcut.

Click the shortcut with the right mouse button, choose Properties, and on the Program tab, select the Close on Exit option. Click Advanced, select MS-DOS mode, and deselect Warn before entering MS-DOS mode. Click OK twice.

Are you ready to restart Windows 95? Double-click your new desktop icon!

Windows97 or Windows98 - Your guess is as good as mine!
Microsoft has decided on the product name for the next version of the Microsoft Windows operating system - Windows9x. Although we don't know if the 'x' will end up as a 7 or an 8. It was previously code named Memphis.

Microsoft is rumoured to be working a couple of new features including:

- Internet/intranet browsing capabilities (Internet Explorer will be included with the final product.)
- Support for state-of-the-art hardware, including the Universal Serial Bus (USB), Digital Video Disc (DVD), and Advanced Configuration and Power Interface (ACPI).
- Support for all the latest multimedia components such as Intel's highly publicised MMX multimedia processor
- Technologies to help reduce the cost of owning and maintaining PCs
- All of the same features - including FAT 32 support for large disc drives - now available in the Windows 95 OEM-only Service Release.

People converting from Windows 3.1 will still have the option to stay with a FAT 16 file system by using a built-in conversion utility. Recovering is not always easy with ERU.

If you use the Emergency Recovery Utility found on the Windows 95 CD-ROM to create an emergency boot disk, be aware that some of the configuration files can be huge, making it impossible to fit them all on a floppy (and ERU can't handle multiple floppies). When ERU shows the files it will back up, click Custom to see the files it will skip. You can choose to skip other files, or you can make a note of the ones not being copied and back them up manually.

Calculated Fields and Totals in Ad Hoc Reports
Calculated fields can't be totalled in O7.3.x ad hoc reports as they did in O7.1.x. The Totals section will incorrectly print the last value of the calculated field.
Workaround - Add a second calculated field e.g. Calc2. Set its calculation to Calc1+Calc2. This produces a running total of Calc1 and hence gives the total of the calculated field, Calc1, in its last value - value which appears in the Totals section.
How do we stop these running totals from printing? Set their text colour to white.
John Froude
john_froude@MSN.COM

A Serialisation Script for Install WISE

```
register.scr:

PROCEDURE ("Startup")

;;Set(destination,"C:\BENCHMRK\OMNIS7.EXE") will be
inserted above by wise
        do("Serialize")
  on (error)
        message("ERROR: Could not complete
Registration!")
  on (break)
        message("Registration was interrupted...")
ENDPROC

PROCEDURE ("Serialize")
        Dialog(dlgSerial)
        Serialise("%O")
        Message("Registration completed!")
ENDPROC
```

prior to running the script, Wise modifies the file by insert a line at line 2 that contains set(destination,"%MAINDIR%\OMNIS7.EXE") this ensures that the script knows where the user has put the OMNIS7.exe file.
The end result is that during the install you are prompted for registration details, if you type in the wrong number you are prompted and if for some reason the serialisation doesn't work you are warned.
John Vagg
micromaze@PEG.APC.ORG

What Is "Use Field of Name" all about?
Use fld of name will evaluate the field name in the first part of the calculate statement using that evaluated "name" as the target for the calculation, the example below will probably be clearer than my attempt in english.

```
Calculate %%string as '#S5' Calculate #S5 as 'Original'
Calculate %%string as 'Different' (Use fld() of name) OK
message {#S5 = " [#S5] "} ;;        #s5 will now show
"Different" as its value
```

Useful for stepping thru fields that are numbered sequentially, ie, MILEAGE1,MILEAGE2,MILEAGE3 and so forth.

From the OMNIS Listserver

The OMNIS Advisor

Welcome to the second installment of the OMNIS Advisor. In this article I will discuss a simple architecture to allow an old version of a library to be automatically replaced by a newer version in a network environment. I've referred to this as a 'low-tech CMS'. Obviously this solution contains none of the features or power of the OMNIS CMS, but under certain circumstances it may fit the bill. 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 Firstly ensure that my users are using the correct version of my application and secondly, automatically distribute new versions with a minimum expenditure of time and money?

The Solution

I have built a simple solution that works in the following manner:
1) In my version numbering scheme, the library has a version number consisting of a number with two decimal places followed by a single letter suffix, such as version 1.10b.
2) The version and suffix of the library is hard-coded in procedure zero of STARTUP (STARTUP /0).
3) The latest available version of the library is stored in a record on the server. By 'server' I mean either a native OMNIS datafile OR a SQL compliant back end database.
4) When a new version of the library is available, I place it in a pre-determined location on the network file server.
5) When a new version is available but it is not required that all users switch to it, I change the suffix but keep the version number the same...for example, I change from 1.10b to 1.10c.
6) When a new version is available and I require all users to switch to it, I change the version number...for example, I change from 1.10b to 1.11a.

What Really Happens

When I've made revisions to my library and I'm ready to release it, I first make sure that the new version number and suffix are correctly hard-coded in STARTUP. I then place the new version of the library on the server, and I update the database to reflect a new 'current version'. From this point forward, when a user runs the application, they will be prompted to upgrade to a new version. If they proceed with the upgrade, their old version of the library will be replaced by the version on the server.

Test the Demo

To best understand this discussion, download the JPCRDEMO.LBR from the following URLs:
Windows:
ftp://ftp.crocker.com/pub/users/pistrang/JPCRDEMO.zip
Mac:
ftp://ftp.crocker.com/pub/users/pistrang/JPCRDEMO.sea.bin

The file that you download contains a read-me (this article, and two libraries, named JPCRDEMO and NEWVER. When you open the JPCRDEMO application, nothing much will happen, since it's currently set to not request an upgrade. To test it, try the following:
1) Keep a copy of the libraries safe somewhere as a backup, and then place two copies of JPCRDEMO in two different directories on your computer. Call one directory Server and one directory Client.
2) Place the NEWVER library in the Client directory.
3) Open the copy in Server, and modify the version number and prefix at the top of STARTUP/0...set it to 1.11a.
4) Open the copy in Client, and modify the fields that contain the hard-coded path name of the location of the Server directory. These can be found in STARTUP/499.
5) Quit OMNIS and reopen the library in Client. Following the prompt, the old version will be replaced with the new one.

Under the Hood

I've loaded the code with comments, so an OMNIS developer should be able to follow what's happening starting with STARTUP/0. Here are a few additional comments:

The NEWVER library performs an important function. My goal, remember, is to replace the JPCRDEMO library with a newer version, while staying in OMNIS and staying connected to the database. To accomplish this, I first open and call into a 'shell' library called NEWVER. NEWVER then deletes the old version of JPCRDEMO (with a Deletefile external command), copies the new version from the server to the client, calls back in to the new version of JPCRDEMO and then closes NEWVER.

The messages and actions in my demo are based on the version numbering scheme described above. If the new version requires a change and the user says 'no' or the server can't be located, we quit OMNIS.

I use a memory-only file format (MOFF) called gv to hold information that I need throughout my application. When I switch from the old to the new version of my library I have already connected to the database server, and I already have information that I need in gv (such as the user ID). I keep the info by creating and defining a list that holds the contents of gv. I pass this list to NEWVER, then pass it back to JPCRDEMO when the new version is available.

Next Column

There have been some threads on the list lately concerning parameters and prompt windows. In my next column and demo I'll discuss using parameters to create some generic prompt windows, which can be called from anywhere within an application and return dates, numbers, and text.

Jim Pistrang
Jim Pistrang is the president of JP Computer Resources, based in Amherst, Massachusetts. He's a full time OMNIS developer, a Certified OMNIS Software Professional, an OMNIS Ambassador, and the co-director of the Boston OMNIS User Group. Visit him on the web at http://www.crocker.com/~pistrang/. or email jim@jpcr.com

Reversible Blocks

During the Prometheus beta tests, I noted some confusion about how reversible blocks work among both developers and at least one US Blyth employee. Considering that these experienced people had difficulty with this concept, I am certain that the relative newcomer could use some help understanding this powerful feature of OMNIS. Here is a detailed explanation that could benefit a great many OMNIS programmers. (Please substitute the word "method" for the word "procedure" if you are using OMNIS Studio.)

Some Basic Definitions
Parts of what I am about to tell you don't appear in the manuals. In fact, I must assume some internal constructs of OMNIS and give names to them to even explain how reversible blocks work.

Let's begin with the concept of a *block* of commands. This is a contiguous collection of command lines that is treated as a unit in some way. A *conditional block*, for example, is a group of command lines enclosed by an *If…* command and an *End if* command. The block, as a whole, is executed if the original condition is met. A *repetitive block* is a collection of procedure lines bounded by *Repeat* and *Until…* commands and is executed repetitively until the exit condition is met.

A *reversible block*, then, is a collection of procedure lines, bounded by the commands *Begin reversible block* and *End reversible block*, that will be "reversed" in some way. It takes the following form:

```
Begin reversible block
        ;block of command lines
End reversible block
;the rest of the procedure
```

We now need to understand what that "reversion" process entails — what it means to "reverse" a command, how OMNIS knows what to reverse, and when and how this happens.

What Can Be Reversed
Let us distinguish between *states of the application* and *actions performed by the application*. A state of the application is a setting of some sort entirely contained in RAM. Examples of states include:
- the value of a variable
- the Main File setting
- the current report, search, output path, etc.
- which record is in the CRB for a specific File
- the installation state of a Menu or Window Format

There are many more of these. In general, such states can be reversed using reversible blocks.
Actions of an application include:
- the printing of a report
- the updating of CRB records to the datafile
- procedural modification of Formats of the application
- operating system actions initiated by OMNIS

These actions are made permanent by reaching outside the RAM partition of the OMNIS application and affecting the "real" world. The printing of a report cannot be reversed, for example, because we can't suck the ink back off the page.

If there is ever any confusion about which commands can be reversed, the Reference Manual's section on Commands is an excellent source of information. In the heading for each command listed, the Manual indicates whether or not that command is reversible.

The State Stack
I must infer this next bit of reversible block lore from knowledge of computer systems in general and observation of OMNIS in particular. There is an area of RAM set aside by OMNIS to track those items that must be reversed. It operates as a *stack* — that is, items are placed on it and removed from it in LIFO (Last In First Out) order, like a stack of dishes in a cafeteria dinner plate dispenser. Since it holds states of the application (and since it is never mentioned in any official manual), let us name it the *State Stack*.

Protecting the Environment
When the command *Begin reversible block* is executed, OMNIS begins loading states of the application onto the State Stack for each reversible command it subsequently encounters. Knowing *what* it loads, though, is key to your understanding of how to use reversible blocks. OMNIS does *not* load the command being executed onto this stack. Rather, it loads *the state that is about to be changed* onto the stack for later retrieval.

For example, if you issue a *Set main file* command, OMNIS loads the current main file setting (the one that is being changed by the execution of this command) onto the State Stack for later retrieval. Then, no matter how many times the Main File setting changes during the course of this procedure, the Main File will return to its *original* setting once the procedure finishes. For further clarity, if the Main File had been set to File A before the reversible block was executed and the Main File is now set to File B within the reversible block, the fact that File A was the Main File is placed on the State Stack.

As another example, suppose we execute the *Install menu* command in a reversible block. OMNIS will load the prior installation state of this menu onto the State Stack. When this command is reversed, the menu is not necessarily removed from the menu bar. It is only removed if it was not present on the menu bar before the *Install menu* command was issued in the reversible block. If it were already on the menu bar when that command was issued, it would remain there after reversal.

Reversible blocks do not perform the opposite action of a command. They restore the state that was *potentially changed* by that command. Consider this a means of "protecting the environment" within your code — a way of automatically returning to a known state. With regard to the protected states, it is as though the procedure had never been run once the procedure is finished. To do this without using the facility of reversible blocks would require that you set up local variables for each state needing protection and then supply code *at each exit point* to restore those states. This could become extremely cumbersome.

Restoring the Environment
Among those who recommend avoiding reversible blocks, there seems to be a general lack of understanding of when the reversal takes place. This appears to have been fostered by some of the early examples (in the OMNIS 5 days) of reversible blocks where an entire procedure would be enclosed in the block, but some non-reversible

action would be performed by the procedure. This led to the false appearance that the final block delimiting command (*End reversible block*) was the reversal point. That is not how it works.

OMNIS pops prior states off the State Stack and restores them at the *exit points* of a procedure — that is, when the procedure terminates execution at its end or at a *Quit* command — *not* when it encounters the *End reversible block* command. The command *End reversible block* should be translated simply as "stop loading items onto the state stack" — it does not "execute" anything in the common sense. If a *Quit* command is encountered before the *End reversible block*, everything is reversed at that point.

Subroutines

If another procedure is called as a subroutine from within a procedure that has executed a reversible block, the subroutine can have its own reversible block with its own partition on the State Stack. When the subroutine is completed and execution returns to the calling procedure, the "protected" states for the subroutine are restored leaving those for the calling procedure (and the entire calling chain leading to that procedure) on the stack. Create the following procedures and execute the first one to see how this works.

```
Main procedure
Begin reversible block
        Calculate #S1 as "Main Procedure"
End reversible block
OK message {[#S1]}
Call procedure First subroutine
OK message {[#S1]}

First subroutine
Begin reversible block
        Calculate #S1 as "First Level Subroutine"
End reversible block
OK message {[#S1]}
Call procedure Second subroutine
OK message {[#S1]}

Second subroutine
Begin reversible block
        Calculate #S1 as "Second Level Subroutine"
End reversible block
OK message {[#S1]}
```

Items placed on the State Stack, then, are "tagged" with the procedure to which they belong so that OMNIS knows what to restore when. If a *Quit all procedures* command is executed at some subroutine level, OMNIS will obligingly restore all states protected at all levels of the procedure stack. It does this in LIFO order as you might expect. It is all very predictable.

Window Initialization

There is one exception to the rule that protected states are restored at the exit points of a procedure. If a reversible block is executed in the initialization procedure for a Window Format (procedure 0 in OMNIS 7 or $construct() in OMNIS Studio), the protected states are restored when the window (instance) is closed. The reversible block in this special case is setting up a protected local environment for the working life of the window.

If you use a modular programming approach where each Window Format contains all the code required to perform any operation on that window, including opening subordinate windows that always stay on top of the main one, this method works wonderfully. If, on the other hand, your programming taste tends toward having many windows doing unrelated things open simultaneously, each able to come to the top and be closed, you will not want to use reversible blocks in the setup procedures of your windows because you have bypassed the predictable nature of this special case use. You could have a similar problem creating multiple simultaneous instances of a Window Class in OMNIS Studio if you allow instances to be brought to the top out of creation order. You simply have to remain aware of the complications you can cause the more exotic your programming style becomes.

Placement of the Block

I generally place the reversible block for a procedure at or near the beginning of that procedure. Since the purpose of the block is to set up a protected environment for that procedure, the beginning of the procedure seems like an appropriate place. The only things I might place ahead of a reversible block are the following:
Help statements in OMNIS 7 *must* be the first command line(s) in a procedure. I can't put a *Begin reversible block* command ahead of them and expect them to do their job. These are no longer an issue in OMNIS Studio, as the help text is now an attribute of an object. Conditional statements that determine whether the procedure should be executed ought to be placed before a reversible block. For example, in a procedure designed to build a list of invoices for the current customer, it might be wise to test whether a customer record is in the Current Record Buffer before going any further. If conditions are not right for executing that procedure, there is no sense in loading the State Stack just to pop items off it again.

There may be some setup commands that I don't want to be reversed and that don't rely upon setup commands I *do* need to be reversed. For example, I may need some *Parameter* statements at the beginning of my procedure in OMNIS 7 (not needed in OMNIS Studio) or I may wish to perform *Set search as calculation* (which is not reversible). It doesn't matter whether these items are performed before or after the reversible block (unless commands within the reversible block depend upon *them*), but I *might* put them before the reversible block.

Clear Procedure Stack

Another command that causes states to be restored from the State Stack is *Clear procedure stack*. Since this command is too often overused, I thought I should give you fair warning here. The purpose of the *Clear procedure stack* command is to cause OMNIS to "forget" the subroutine return path and treat the currently executing procedure as thought it were the primary procedure (the one initiated by some direct action of the operator). It does not clear any other memory buffers or perform RAM cleanup operations as its name might suggest. Its best use, in my experience, is in halting all procedures if you have been kicked out to the debugger while testing your application. In this case, it can be selected from the Stack menu in the debugger overlay. To protect the environments you have placed on the State Stack, OMNIS pops *everything* off the State Stack when you issue *Clear procedure stack* as though you had issued a *Quit all procedures* command. It does not affect State Stack contents put there by window initialization procedures.

Multiple Reversible Blocks

It is quite possible to have more than one reversible block in a procedure and there are many reasons for doing so. This is done

when you need to add more states to the protected group for the current procedure. For example, suppose we have a procedure that is designed to insert a new Invoice record with associated Line Item records from a single data entry pass using a table field on a window. We will need to change the read/write state of certain Files a couple of times along the way. We assume here that all Files are set to read-only by default and we want them returned to that state when the procedure finishes. Here is one possible method using multiple reversible blocks:

```
Begin reversible block
        Set main file Invoice file
        Set read/write files {Invoice file,Customer
file,System file}
End reversible block
;data entry and insert-update of Invoice record
Set main file Line items file
Set read-only files {Invoice file,Customer file,System file}
Begin reversible block
        Set read/write files {Inventory file,Line items file}
End reversible block
;create line item records
```

Note that the second reversible block did not need to contain the second *Set main file* command or the command to change the read/write mode of the first three File Formats. Alternatively, I could have simply set the Inventory File and the Line Items File to read-only mode in the first reversible block and not needed the second. That would not have *changed* the mode of those two Files, but it would have put their prior state onto the State Stack for later retrieval.

Nested Reversible Blocks

On the other hand, it doesn't make sense to *nest* reversible blocks in the same way we might nest conditional or repetitive blocks. Nested *If…* statements make sense in that we are testing a further condition and performing some action based upon that additional condition. The conditions in the *If…* statements are different and the inner condition is completed and no longer an issue once the corresponding *End if* command is executed.

In a similar way, nested *Repeat* statements make sense because the exit criteria in their separate *Until…* statements are different for each loop.

Since an *End reversible block* command does not perform the reversion, nested *Begin reversible block* commands don't really make any sense. Encountering a command that means "start putting items on the State Stack" when you are already doing so is somewhat useless. The effect the unititiated might hope to achieve by this nesting — protecting some states, performing some action, and then restoring the original states — is accomplished by calling a subroutine with its own reversible block. OMNIS does not keep you from nesting reversible blocks — but it also allows you to comment comments. The effect is about the same.

$construct() and $destruct()

The issue that originally prompted this article involved the new way that windows are initialized in OMNIS Studio and the way we can now control what happens when they close. We now have a method named $construct() that is used to initialize a window instance *before it opens* (procedure 0 in OMNIS 7 is executed *after* the window has opened, requiring a redraw of that window if anything in that procedure changes what the window should display). With respect to reversible blocks, it operates just like procedure 0 does in OMNIS 7.

We also have a method named $destruct() that comes into play just before a window closes. The question one beta tester had was why we should bother with reversible blocks in this method when we could simply undo everything in $destruct(). His specific question concerned mounting a menu reversibly in $construct(). The easy answer was that it didn't matter, that mounting the menu reversibly was the same as removing it using $destruct() — and it was further implied that $destruct() would be a better choice.

But that is not exactly true, as we have seen in this article. We would have to set up a class variable in $construct() to store whether that menu were already on the menu bar as the window was instantiated. An instance variable would be better (in case we open multiple instances of this Window Class), but instance variables don't exist until the instance exists. Our $destruct() method would then have to test this variable to determine whether or not to remove the menu. Using a reversible block in $construct() to mount the menu requires less work on our part as the proper disposition of the menu is dealt with automatically — although we still have to test to see if an instance of that menu exists in $construct(), if we're going to be thorough, so we don't end up with two instances of the menu!

Fear of the Automatic

An argument I have heard from some developers for not using reversible blocks is that they don't like (or trust) things that are automatically done for them by the programming language. Another is that it's just not the "standard" way to do things in an object-oriented environment. While I can understand the desire to maintain control in my programming, I find that reversible blocks actually help me maintain control while using much less code that the "muscle" method. Based on such an argument, one might also abandon use of local variables, since OMNIS automatically destroys their memory allocations at the end of a procedure — outside our control.

As far as "standards" are concerned, if OMNIS has a *better* way, then use it. Let's hope that OMNIS isn't always playing "follow the leader"! Just because other products don't have some of the features of OMNIS doesn't mean those features are bad. (Perhaps we'd be better off using the "leader"…)

As we have seen here, reversible blocks perform their function in a very predictable way. We have complete control over what gets reversed and when, as long as we follow some basic structure in our application design.

Things we don't understand are often scary. Hopefully, this article has demystified reversible blocks a bit for you and made them more approachable. Reversible blocks are very easy to use once you understand them and they can become an important design element that will save you time and effort in your OMNIS programming career.

David Swain
David Swain, owner of Polymath Business Systems in Bedford, New Hampshire has been teaching OMNIS to eager students around the US since early 1985. His technical journal, OMNIScience, remains a great source of detailed information on important features of OMNIS. You can learn more about these offerings through his Web site at http://www.polymath-bus-sys.com/.

Fast Import and Export of Lists on the Mac

If you are running OMNIS 7 on a Macintosh, there is a very fast way to export/import text to a list. Using AppleEvents, OMNIS can issue a "Set Data" command to itself which will translate a list to a text field or other other way around. The list turns into a TAB delimited text string. The code is very simple:

Enable receiving of Apple events Set event recipient Send Core event {Set Data ('#S1',#L1)} Disable receiving of Apple events

The receiving field must be in quotes and the sending field not in quotes.

Contributed by: Fred Haislmaier
fredh@EFCAH.com

PowerPrinter Copyright (c) 1997 By DigitalWave

PowerPrinter is a 32Bit Windows DLL that gives you total control over the printer. PowerPrinter is a brand new 32Bit DLL designed to support both Windows 95 and NT (3.51 and 4.0) shielding the developer from the OS differences. However, PowerPrinter allows as well the developer to take advantage of features unique to each OS such as Windows 95 ICM support (planned for v1.1).

PowerPrinter can be used from any development tool (e.g. PowerBuilder, Visual Basic, Delphi, Centura, OMNIS.) capable of calling external functions.

Pricing: US$ 35 per developer. No runtime royalties/fees. Detailed ordering information can be found in the help file.

Upgrades: free for registered customers

Support: free by email and fax (+352.305.955)

Contact:
support@digitalw.com on the Internet or
100441,1136 on CompuServe
Web: http://www.digitalw.com
DigitalWave Limited 35, Rue du Kiem L-8328 Capellen
Luxembourg

PowerPrinter v1.0 Functions supported
dwAbout

Constructor / Destructor
CWin32Prn_CPP_CONSTRUCTOR()
CWin32Prn_CPP_DESTRUCTOR

Unlock (registered users only). It disables the NAG Screen.
dwUnlock,dwGetDefaultPrinterName,
dwGetDefaultPrinterPort, dwGetDefaultPrinterDriver,
dwSetDefaultPrinter, dwGetPrinterList,
dwGetDefaultPrinterEx, dwSetDefaultPrinterEx

dwGetPrinterOrientation, dwSetPrinterOrientation,
dwGetPaperSize

dwSetPaperSize, dwGetSupportedPaperSizeList,
dwGetPaperLength, dwSetPaperLengthd,
wGetPaperWidth, dwSetPaperWidth, dwGetScale
dwSetScale, dwGetCopies, dwSetCopies,
dwGetDefaultSource, dwSetDefaultSource,
dwGetPaperBinListm, dwGetNamedPaperBinList

dwGetPrintQuality, dwSetPrintQuality, dwGetColor,
dwSetColor, dwGetDuplex, dwSetDuplex,
dwGetYResolution, dwSetYResolution, dwGetTTOption,
dwSetTTOption, dwGetCollate, dwSetCollate

Printer Access dwGetPrinterAccess, dwSetPrinterAccess

Once you have downloaded the demo from their web site you will need some code to get you started as you need the handle to the dll in memory. The code below describes how to do this and provides some simple examples.

1 Get Handle —————Calculate #1 as RegisterDLL ("KERNEL32","LoadLibraryA","JC") CallDLL ("KERNEL32",'LoadLibraryA',"C:\Win95\System\powerprn.dll") with return value #1 ; Full path to dll is not necessary if in the same directory as OMNIS or In the WIndows System directory Calculate #1 as RegisterDLL ('powerprn.dll','CWin32Prn_CPP_CONSTRUCTOR','J') CallDLL ('powerprn.dll','CWin32Prn_CPP_CONSTRUCTOR') with return value #1

2 About ————RegisterDLL ('powerprn.dll','dwAbout','V') CallDLL ('powerprn.dll','dwAbout')

3 Default Printer Driver ————————— RegisterDLL ('powerprn.dll','dwGetDefaultPrinterDriver','JJC') CallDLL ('powerprn.dll','dwGetDefaultPrinterDriver',#1,#S1) with return value #2 OK message {Default Driver [#S1]}

4 Default Printer Name ———————— RegisterDLL ('powerprn.dll','dwGetDefaultPrinterName','JJC') CallDLL ('powerprn.dll','dwGetDefaultPrinterName',#1,#S1) with return value #2 OK message {Default Printer Name [#S1]}

5 Default Printer Port ————————RegisterDLL ('powerprn.dll','dwGetDefaultPrinterPort','JJC') CallDLL ('powerprn.dll','dwGetDefaultPrinterPort',#1,#S1) with return value #2 OK message {Port [#S1]}

6 Default Printer List ————————RegisterDLL ('powerprn.dll','dwGetPrinterList','JJC') CallDLL ('powerprn.dll','dwGetPrinterList',#1,#S1) with return value #2 OK message {Printers [#S1]}

7 Unload Dll —————RegisterDLL ("KERNEL32","FreeLibraryA","JJ") CallDLL ("KERNEL32","FreeLibraryA",#1) with return value #2

Contributed by:
stephen_miller@dlagroup.com.au

Microsoft ActiveX

With the imminent release or OMNIS Studio a working knowledge of ActiveX Controls will be critical. The following short article from MicroSoft sets out the benefits of the ActiveX Technology. Reference is made to the development of ActiveX for the MacOS with Metrowerk's CodeWarrior which may seem curious to some readers however it is also rumoured that the 'Office' Suite for the Mac is currently being rewritten in CodeWarrior so it would appear that a significant relatioship has developed between the two groups.

ActiveX

What is ActiveX?

ActiveX is a set of technologies from Microsoft that enables interactive content for the World Wide Web. With ActiveX, Web sites come alive using multimedia effects, interactive objects, and sophisticated applications that create a user experience comparable to that of high-quality CD-ROM titles. ActiveX provides the glue that ties together a wide assortment of technology building blocks to enable these "active" Web sites.

What Are Its Primary Benefits?

❒ Active Web Content with Impact that will attract and retain users.
❒ Open, Cross-Platform Support on Macintosh, Windows and UNIX operating systems.
❒ Familiar Tools from a wide assortment of tools and programming language vendors, including Visual Basic, Visual C++, Borland Delphi,
Borland C++, Java, and Java-enabled tools. Developers can use what they know and be productive immediately.
❒ Existing Inventory of ActiveX controls available today for immediate use by Web producers.
❒ Industry Standards, with built-in support for key industry and de-facto marketplace standards, including HTML, TCP/IP, Java, COM, and others.

What Are Its Elements?
ActiveX includes both client and server technologies.

❒ ActiveX Controls are the interactive objects in a Web page that provide interactive and user-controllable functions and hence enliven the experience of a Web site.
❒ ActiveX Documents enable users to view non-HTML documents, such as Microsoft Excel or Word files, through a Web browser.
❒ Active Scripting controls the integrated behavior of several ActiveX controls and/or Java Applets from the browser or server.
❒ Java Virtual Machine is the code that enables any ActiveX-supported browser such as Internet Explorer 3.0 to run Java applets and to integrate Java applets with ActiveX controls.
❒ ActiveX Server Framework provides a number of Web server-based functions such as security, database access, and others.

What Can It Do?
ActiveX brings innovation and interactivity to the Web. Because it is supported by many different languages and tools, it enables developers with varied backgrounds and expertise to bring their creativity to the Web. Based on a refinement of the existing COM standard already known by thousands of developers, it can leverage the knowledge and work of the development community without a steep learning curve. And because it is a third-generation technology with extensive third-party support, it provides the richest development platform for both Internet and intranet Client/Server applications available today. ActiveX takes the most creative and innovative software development efforts and enables them to work together seamlessly in a Web site. With thousands of these software components already existing, an exciting collection of interactive objects is available for immediate use by Web producers.

Why Is It Important?
ActiveX makes it fast and easy for developers and Web producers to create unique, interactive Web sites that will make the Internet fundamentally more useful and productive. Web producers don't have to start from scratch and build all the parts of their interactive Web site by hand, because there are already more than 1,000 reusable controls available in the market. And because ActiveX can be used with a wide variety of programming languages from dozens of vendors, developers and Webmasters can make use of their current expertise to more quickly create compelling content. They can also accommodate a wide range of users, as ActiveX will be supported on multiple operating system platforms.

How Does It Compare with Java?
ActiveX provides a standard mechanism to extend any programming language, including Java. ActiveX extends the capabilities of the Java language by allowing Java developers to integrate their applets with the richness of ActiveX. ActiveX ties Java applets together with objects created in other languages, so that Java programmers can link to ActiveX controls directly from their Java programs. By the same token, objects written in other programming languages from multiple vendors can link to Java applets. ActiveX is the glue that ties them all together, delivering the most powerful Web technologies in an open, integrated platform. By providing a common way to extend and link programming languages including Java, ActiveX maximizes developers' resources for interactive Web development. See Java and ActiveX for more information on extending Java with ActiveX.
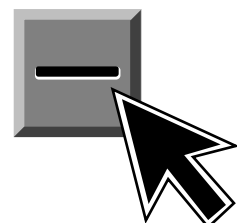
Who Supports It?
Small, medium and large software companies currently create ActiveX controls, including companies such as Borland, Oracle and Sybase/Powersoft. As a result of their work, there are more than 1,000 existing ActiveX controls available for use today by Web producers. In addition, 14 companies who create Web design and development tools have built ActiveX support into their products, allowing their customers to both create and make use of ActiveX controls in their programs. Microsoft's Internet Explorer supports ActiveX, and Microsoft provides the ActiveX plug-in for Netscape¨ Navigator¨, enabling the broadest range of Internet users to view ActiveX-enabled Web pages.

Where Does It Run?
ActiveX is currently supported on the Windows operating system. Microsoft is working with Metrowerks to support ActiveX on the Macintosh platform, and is also working with Bristol and Mainsoft to support it on UNIX platforms. Developers who write ActiveX controls and other ActiveX objects will be able to reach the widest possible user audience with this cross-platform solution.

OMNIS®
Web
Enabler

This exciting new tool is available to Australasian OMNIS developers at A$799. Send all orders to Amanda Turner <amanda_turner@dlagroup.com.au> or fax her on +612-9262-2290.

ONLY A$799

● Macintosh FTP application with files

● OMNIS can perform as a Web Server

● Windows screen shot of Email application

OMNIS®

Name:_____

Company:_____

Tel:_____Fax:_____

I enclose Cheque/Money order for:_____

Please Specify:　Mac ❏　or Windows ❏

Bankcard❏　Visa❏　Mastercard❏　Amex❏

Card No:|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Expiry Date:　/　/

Cardholders Signature:_____

Please return completed form to DLA Technology,
Level 3, 35 Clarence Street, Sydney NSW 2000　Tel: (+61 2) 9262-2255 Fax: (+61 2) 9262-2290

AHEAD OF THE CURVE

OMNIS

software

inc.