

## Novell Visual AppBuilder

### White Paper

July 1993

Since the first computer was envisioned, computer programmers have played the role of problem solvers. Usually, computer users in a large company communicate their needs to in-house developers, who then create applications to solve the specified business problems. Although developers work closely with the users to understand their needs and provide the appropriate solutions, users often find themselves with an application that does not completely meet their requirements. Frequently, by the time the developer finishes writing the application, the business need has changed.

Medium to small companies play out this same scenario with off-the-shelf software. Rather than having a development team on staff, the organization relies on an independent software vendor's (ISV's) attempts to anticipate users' needs and write an application that addresses them. The result is broadly written, general software that offers a company no competitive edge.

Writing applications requires specialized skills such as knowledge of programming languages, syntax and operating systems. These skill requirements have limited the number of people who can successfully create today's business applications. Novell recognizes the backlog of network applications and in response has designed AppWare, a system for developing network applications. The Novell Visual AppBuilder, one of the components of AppWare, enables business application developers to construct applications without programming or system knowledge.

### **What Is Novell Visual AppBuilder?**

Novell's Visual AppBuilder is a high-level programming tool based on technology Novell acquired with the purchase of Serius Corporation. Tightly integrated with AppWare, Novell's Visual AppBuilder allows developers to graphically design and implement the business solutions users need to be productive — without writing a single line of code or syntax.

Visual AppBuilder gives business application developers the ability to leverage their computing environment to gain a strategic advantage. It empowers them to solve problems efficiently and effectively, without having to worry about the complexities or inner-workings of a heterogeneous computing system. In fact, business application developers have no need to learn the topology of the network, the platform on which their application will be implemented, or whether the services being used are provided by mainframes or PCs. The result is a more productive workplace.

## **AppWare Loadable Modules (ALMs)**

Visual AppBuilder enables the business application developer to quickly build portable software applications by linking software components called AppWare Loadable Modules (ALMs). An ALM is composed of an object and its associated functions. ALMs can range from simple graphic utilities, spreadsheet modules, multimedia engines and structured query language (SQL) client-server connections, to services provided by NetWare and other network platforms, such as file, print, electronic mail, imaging, telephony and directory services.

Applications created with Visual AppBuilder are built on the AppWare Bus technology, a software engine that manages and coordinates the interaction of ALMs (see Figure 1 on page 3 of the hard copy). The AppWare Bus allows an application to access the functionality provided by ALMs and it enables the ALMs to communicate with each other. Each ALM's object and function is represented by icons. Developers do not have to know whether a service is located in a local operating system or at the network level, or how these services interoperate. Instead, an application is simply constructed by combining different ALM icons. Just as people purchase separate stereo and home entertainment components to build a stereo system that accommodates their specific needs and tastes, Visual AppBuilder interfaces different ALMs to build an application that meets specific business objectives. This makes Visual AppBuilder an ideal tool for developing applications that keep pace with today's rapidly changing business environment. For example, a marketing department now can build a multimedia presentation using sound, video and imaging ALMs.

## **Building and Using ALMs**

Novell provides developers the necessary tools for creating ALMs. These tools include:

- The ALM Construction Kit, which provides the interfaces necessary to support the AppWare Bus's object interaction protocol (OIP). OIP is the protocol for passing information and flow of control from applications to ALM components.
- The AppWare Foundation, which provides a consistent set of APIs that allows developers to access the services provided by both the network and local operating systems.

Using the ALM Construction Kit and the AppWare Foundation, developers can write and compile ALMs using 3GLs, including C,

COBOL and Pascal. Novell is actively working with a number of third-party vendors to ensure the availability of a wide variety of ALMs.

Instead of trying to constantly adjust to changing user needs, traditional developers can now concentrate on creating general purpose ALMs that deliver customizable computing and networking services that extend the AppWare environment. Business application developers throughout an organization can then use these ALMs to build applications that solve specific business needs. For example, a banking organization could have three ALMs to provide key functionalities: a loan ALM, a daily deposit ALM and a certificate of deposit ALM. These ALMs could be used to build applications that allow the loan department to perform loan transactions.

ISVs can follow the same line. Instead of building an off-the-shelf application that attempts to meet all customers' needs, they can build ALMs and let companies create the customized applications they desire.

## **Desktop Portability**

A major benefit of Visual AppBuilder applications is their portability between desktop platforms. A recent Forrester Research report indicates that companies are currently facing the challenge of a heterogeneous computing environment and therefore need a heterogeneous solution. ALM functionality extends across multiple platforms because it takes advantage of the AppWare Foundation, which spans both local operating systems and distributed services provided by the network (see Figure 1 on page 3 of the hard copy). For example, an application may access a file on the local data storage disk with one read; on a subsequent read, it may access a file on a network server. Since the AppWare Foundation provides Visual AppBuilder with one consistent interface, developers no longer have to worry about platform compatibility.

For example, projects built with Visual AppBuilder for Macintosh can be recompiled as an MS Windows application using Visual AppBuilder for Windows, and vice versa. Porting projects between Macintosh and MS Windows platforms becomes seamless and virtually instantaneous. Visual AppBuilder includes a core set of ALMs supporting all aspects of the Macintosh and MS Windows graphical user interfaces (GUIs), such as menus, windows, window items/controls, window items/fields, window items/pictures, cursors and keyboard. Operations include application operations, repeating operations, manipulating numbers and text, grouping ALMs, alerts, file manipulation, printer setup and control, and a full-function database ALM for building multiuser,

relational databases.

## **Building Applications with Visual AppBuilder**

Applications are built with Visual AppBuilder by connecting graphical icons representing ALM objects and functions to indicate sequences of operations and data flow (see Figure 2 on page 5 of the hard copy). ALMs can be connected in almost unlimited combinations to create programs more quickly and cost-effectively than currently possible through the use of conventional programming tools or even 4GLs. Applications can be modified easily by removing ALMs or adding new ALMs as they become available.

### **Current ALM Sets**

**Serial Communication** - For custom communications software, such as service bureaus, bulletin boards and terminal emulation. This set includes basic connectivity tools such as modem, serial, ADSP, TTY, VT102, text and XMODEM. The Connection ALM in this set facilitates serial communication with a variety of peripheral devices. Additional communication tools are available from third parties.

**InterApp Communication** — For custom electronic mail, office automation, network administration and application scripting and control. ALMs in this set enable applications to send and receive text, pictures, numbers, sounds and other data.

**Multimedia** - For presentation, interactive training, education and multimedia databases. ALMs in this set support animation, QuickTime movies and sound.

**XMCD** - Provides the ability to incorporate HyperCard XCMDs and XFCNs. This ALM set provides the most powerful third-party XCMD extension available, and supports the original HyperCard XCMD format, as well as the extended XCMD interface for HyperCard 2.0 XCMDs.

**SQL** - Provides various client-server SQL access including Oracle, Sybase, Btrieve and Data Access Manager.

### **Debugging Applications**

Visual AppBuilder includes an integrated debugger that helps test the data flow of the application. The tool allows a "breakpoint" or stopping point to be placed on any link connecting ALM objects and functions. When running in delay mode, the application stops at each breakpoint, allowing the developer to examine object contents before and after various functions execute. If the function being examined is at the end

of a chain, dummy functions can be inserted for testing values or field contents.

## **Creating Reusable Subprojects**

Business application developers can create separate subprojects within an application and reuse them by simply copying and pasting them between applications like a picture or paragraph of text. Subprojects reduce workgroup development design time considerably by allowing the developer to focus on a discrete application segment and create subprojects for it. These subprojects can be copied and pasted together into a final design that works as a single application. An example of this is an edit menu with pull-down functions such as copy, cut, paste, delete and insert. Once this edit menu is created, it can be named and reused repeatedly.

## **Facilitating Code Reuse**

While low-level object classes are reusable, working with them requires greater expertise as well as much more effort on the developer's part. ALMs are completely encapsulated or "black-boxed," making them easier to use and reuse. And a Visual AppBuilder user never needs to know how an ALM was implemented in order to use it.

The complete encapsulation that characterizes ALMs makes them easier to update and/or replace. Updating ALMs in a Visual AppBuilder project is as simple as copying an ALM file to the ALM folder or replacing a Dynamic Link Library (DLL) in a directory. No reprogramming need be done. In many cases, no recompilation is necessary.

## **Raising the Level of Application Development**

ALMs eliminate the need for line-by-line programming, which greatly reduces time to market, cost of maintenance and the learning curve required to become proficient with the tool. And Visual AppBuilder's rapid development cycle makes possible a more responsive and direct interaction between developer and user, with consequent benefits in application look and feel.

## **Reducing the Learning Curve**

Conventional programmers must not only master the constructs of given programming languages, but also those of the operating systems, GUIs, and network operating systems and services. Although some basic understanding of programming logic is necessary, at no

point is a Visual AppBuilder programmer required to know anything about the operating system calls required to make ALMs functional.

## **Creating Small, Fast, and Efficient Applications**

All applications built with AppWare are compiled, not interpreted. Code overhead imported from the Visual AppBuilder environment is just 25KB. Furthermore, ALMs take full advantage of the operating systems for which they were designed. As a result, applications are smaller and faster than their counterparts developed in other systems which rely only marginally on operating system services. Further, applications have the standard look and feel of their respective platforms because the underlying ALMs are created using the AppWare Foundation.

## **Porting Applications Across Different Platforms**

Because Visual AppBuilder projects contain no source code, they are easily ported from one platform to another — for example, from MS Windows to Macintosh. Projects are simply symbolic representations that indicate how source code segments will be combined when the project is compiled. Creating cross-platform applications, therefore, involves translating the project file (a comparatively simple task) and then recompiling it on the destination platform using that platform's ALM components. The object code for each platform is designed specifically for that platform, so ported applications look and work like any other software on that platform.

## **Summary**

As a key component of AppWare, Visual AppBuilder enables business application developers to build reusable, extendible, cross-platform applications without having to deal with low-level programming issues. By using Visual AppBuilder, graphical interface and services can be interlaced to build the logic and flow of new applications, representing thousands of lines of code. Using the ALM Construction Kit, a programmer can take existing source code and fashion a reusable ALM that can be shared across platforms.

By leveraging NetWare's rich set of services within a highly productive, rapid application development environment which requires no networking knowledge, Novell continues its steadfast mission to expand the network computing industry.

(c)1993 Novell, Inc. All rights reserved. Novell, the N design, and NetWare are registered trademarks, and AppWare, AppWare Bus, AppWare Foundation, AppWare Loadable Module, and Visual AppBuilder are trademarks of Novell, Inc. FlexOS is a registered trademark of Digital Research, Inc., a wholly

owned subsidiary of Novell, Inc. AppleTalk, HyperCard and Macintosh are registered trademarks, and QuickTime is a trademark of Apple Computer, Inc. DECnet and VAX are trademarks of Digital Equipment Corporation. IBM, AS/400, OS/2, and SAA are registered trademarks, and MVS is a trademark of International Business Machines Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation. Sun workstation is a trademark of Sun Microsystems, Inc.