

# Contents

## Introduction

3

Personal Digital Assistants (PDAs) have momentum—they're being used by a wide mix of organizations and individuals around the world. Newton PDAs are perfect companions for Windows and Mac OS computers. Newton offers developers a host of advantages, including rich built-in functionality, a set of powerful development tools, and strong support and training programs.

## Section I: The Newton PDA advantage in the development process

5

The Newton platform has been built for rapid, flexible application development. At every level of the system, the platform offers functionality that speeds the development process. The Newton operating system is completely object-oriented and provides an advanced recognition architecture and communications architecture. The platform's system services level offers a host of systemwide functions including printing, faxing, routing, and filing. At the component level, Newton 2.0 offers dozens of components that a developer can call on to build an application—buttons, lists, tables, input fields, and so forth. All of this translates to faster development efforts. The platform also offers tools to unlock the power of the platform: Newton Toolkit is a powerful application-development environment that's optimized for the creation of handheld applications. Newton Book Maker and Newton Press enable you to create electronic books and on-line documentation tools. Electronic forms tools provide an easy way to create forms-based systems that connect to the enterprise. And, at every step of the way, Apple provides technical help through a set of programs, resources, classes, and self-help tools.

## Section II: Application development case studies

19

In this section, you can see how Newton tools help speed the development process. We cover four different development teams and their work: The Intuit Pocket Quicken team, a team that developed a handheld application for medical interns, a team that developed a quality-control and inspection tool for a major fast-food chain, and a team that developed a salesforce automation tool for a car salesforce.

# Introduction

Personal Digital Assistants (PDAs) have momentum—they're being used by a wide mix of organizations and individuals around the world.

Forrester Research, an independent firm, forecasts that by the year 2000, the installed base of PDAs will reach 8.2 million in the United States alone. Today, independent software vendors (ISVs) are taking advantage of the Newton platform's built-in functionality, its strong communications capabilities, and its development tools to create unique applications that will help users to organize, communicate, and integrate information.

In-house development teams in companies around the world are creating mission-critical applications that rely on Newton PDAs for a wide variety of uses—from electronic information distributors on hospital floors to salesforce tools for car dealerships; from quality control devices in restaurant chains to sophisticated data collectors for businesspeople; from crime-reporting tools for police officers to communications devices for mobile professionals.

Why are Newton PDAs finding success? Because they offer a host of advantages: They're small and portable, which means users can work with them anytime, anywhere. They're powerful, offering the processing power of an average desktop PC, which means they can perform countless useful tasks in the field. And they're great companions to personal computers—both Windows and Mac OS systems—because they can share and integrate information with desktop applications within the enterprise.

Moreover, on the Newton platform, custom solutions are easy to create. Newton 2.0 offers developers and systems integrators a superior platform for rapid applications development. Here's what the Newton platform has in store for developers:

- **Rich, built-in functionality.** The Newton platform offers development teams a lot on which to capitalize. And they don't have to start from scratch—much of the work has already been done for them. Developers can capitalize on the built-in functions—handwriting recognition, faxing capabilities, printing capabilities, and so forth—in their applications simply by writing a few lines of code. And the platform offers a versatile communications architecture that enables developers to easily share data in applications over a wide variety of means—both wired and wireless.
- **A set of powerful development tools.** The platform is open—and was designed from the ground up for rapid applications development. As demands on handheld applications grow more sophisticated, and as lifecycles of applications become even

shorter, the Newton approach to development makes even more sense. With Newton tools, developers can see the results of their efforts in minutes, and build their applications interactively and incrementally—testing the design as they go along.

- **Strong support and training programs.** Apple offers a range of training and support programs to ensure that development teams make rapid progress in creating applications. And as the most mature PDA platform, Newton has a large number of programmers and systems integrators who can work with you as you build your application.

For these reasons, thousands of developers have made the Newton platform their PDA platform of choice. Simply put, Newton is a sophisticated, mature, robust, and flexible PDA software development platform.

This paper focuses on the advantages of the Newton platform for rapid applications development of horizontal, vertical, and custom applications.



# Section I

The Newton PDA advantage in  
the development process

---

### PDAs: New uses and applications

Because of their portability, strong communications capabilities, and easy-to-use interface, Newton PDAs are enabling mobile professionals to work in entirely new ways, and Newton developers have created a wide variety of tools tailored to meet their needs.

- **Salesforce automation tools** that give sales representatives instant access to product data—as well as the ability to access and share inventory data and ordering information with the home office.
- **Health care tools** that enable doctors, nurses, and other professionals to access critical medical reference information—and even patient information—while on the hospital floor.
- **Field-service applications** that enable personnel to stay in touch with dispatch, learn new procedures in the field, and follow step-by-step on-line instructions.
- **Inventory systems** that let auditors in the field quickly record inventory levels of anything from packaged goods to auto parts.
- **Teaching tools** that guide students through exercises and field activities, enabling them to take reference material into the field—as well as record their findings.
- **Process-control auditing tools** that enable teams to inspect locations—for example, fast-food chains—to ensure that processes are being followed and that safety standards are met.
- **Internet browsing tools**, and tools that enable users to access data over corporate TCP/IP networks.
- **Off-the-shelf applications—a host of them**—that are used for everything from personal information management to finance, from electronic guides to e-mail and messaging systems, from outlining and presentation programs to spreadsheets. (For examples of how the Newton platform is being used in business and education, see Section II of this paper.)

### The challenges of the development process.

Of course, developing sophisticated systems such as the ones listed on page 7 have—until very recently—presented significant challenges. And, certainly, any discussion of software development for handheld devices must first start with a look at the development process itself—a process that presents many of the problems that have always faced developers, as well as a few new ones that are specific to PDA platforms.

- **Graphical user interfaces are not easy to build.** On most handheld platforms, developing a true graphical user interface—and building true ease of use into any application—can require a fair amount of coding.
- **Managing memory is difficult.** Since most handheld devices have limited memory, managing the use of memory is critical to overall application performance. Most platforms present challenges in this area for programmers.
- **Building in functionality and access to services, such as communications and network services, can be difficult.** In many cases, for an application to be truly successful it must operate within the larger context of a corporate network: It must share data with existing databases, take advantage of existing printers, and use existing networks—wired or wireless—to communicate.
- **Data storage and information management can also be daunting.** If they're to be of value to users, handheld devices must be able to deal with a wide variety of information, enabling users to effortlessly capture data on-the-fly, organize it, search it, share it with other users, and transfer it to a personal computer and host system. Information management and integration with personal computers are often key to the successful deployment of a handheld device.
- **Applications maintenance, revising, and upgrading.** For both custom and off-the-shelf applications, the ability to update—to change functionality, add new features, or otherwise improve the way it works—is critical. Traditionally this has been difficult, because applications were so interwoven and complex that adding new functionality required major programming efforts.
- **Communication with personal computers.** Exchanging information, importing data, and integrating with PC applications and e-mail present challenges to the PDA developer—merging the two worlds and two platforms is not a simple matter.
- **Battery life and screen size.** PDAs present some other challenges that aren't common in the personal computer world. For one, since PDAs are portable and rely on batteries for power, it's necessary to think about how power is used—and how to preserve it for long battery life between charges. Second, because their screens are small, you also need to consider how to make the best use of them.

### Newton: Built from the ground up for flexibility, performance, and integration into today's complex environments.

Designed and optimized for small, handheld applications, the Newton platform offers developers an open architecture that directly addresses the challenges listed above.

Say your goal as a development team is to create a handheld application—one that's easy to use and graphical in nature, and that easily shares data with desktop PCs and host systems. And let's say you also want that application to be able to evolve over

time, as you add new capabilities and enhance existing ones—based on user feedback, changing business needs, or shifts in target users.

No problem. The Newton platform is designed for the creation of applications that meet all these criteria. The platform is rooted in an architecture that provides reusable prebuilt elements, as well as a broad range of system-level capabilities and low-level and high-level development tools to design, develop, and debug an application quickly. In other words, a lot of the hard work has already been done for you—so rather than having to spend time creating the fundamentals of an application, you are freed to spend time developing the unique and compelling parts of your own application. Following are just some of the advantages that the Newton platform offers developers.

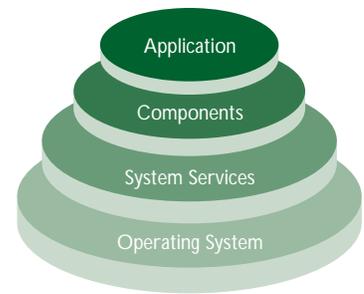
### Built-in functionality: Choosing the Newton platform means never having to start from scratch.

The Newton platform represents a state-of-the-art approach, providing developers with an open, object-oriented architecture that's designed for fast development and code reusability. Current development efforts can capitalize on previous ones, because objects can be called and reused, as appropriate.

At right is a diagram of the Newton architecture. Each level—from the operating system to system services to built-in components in ROM—provides built-in functionality that you can capitalize on in building your application. Let's examine those functions, level-by-level:

#### Operating system

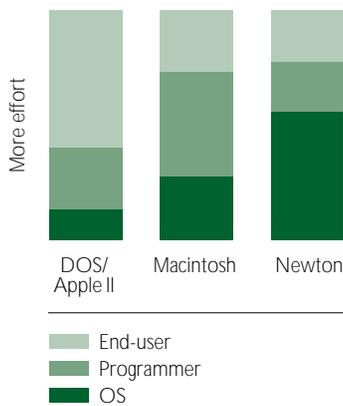
Rather than take a desktop architecture and shrink it down, Apple started from scratch in building the Newton architecture—in order to give you advantages as a developer. The state-of-the-art operating system is built for ease of use and performance.



The Newton platform offers developers a rich set of capabilities at every level.

- The OS itself is completely object-oriented and provides—among other things—an advanced recognition architecture and communications architecture, lessening the developer's programming load for building an application.
- The system services level offers the capability of storing and viewing data, as well as a host of system-wide functions including printing, faxing, routing, and filing.
- At the component level, Newton 2.0 offers more than 150 components that a developer can call on in building an application—buttons, lists, tables, input fields, and so forth.
- The purpose of this architecture, of course, is to greatly reduce the time it takes to develop a Newton application—with each level providing advantages to greatly speed the developer's work.

- **Object-oriented architecture.** As a developer, you will benefit from several advantages, because the architecture is object-oriented to the core. That means that new functionality is much easier to add. It also means that you can make the most of built-in functionality merely by writing code that takes advantage of built-in objects. This makes for faster development efforts, since you have a strong foundation on which to build.
- **Multitasking.** Newton PDAs are multitasking, enabling you to perform system functions—such as printing and faxing—in the background. This makes it possible for you to create flexible applications that truly meet customer needs.
- **Recognition architecture.** The Newton platform offers you a sophisticated recognition architecture that recognizes cursive writing, printed text, drawings, and symbols. Your application can call on these sophisticated recognition engines to enter data. Newton 2.0 offers a much higher degree of accuracy than previous versions of the software, making it possible to build applications that can interpret—with a high level of precision—what a user has written on a Newton PDA screen.
- **Communications architecture.** The Newton platform is built for communications, and the architecture is completely modular. Software modules for different transports are built in, and it's easy to handle new ones. New transports are available to the user—and, as a developer, you don't even have to change your application. The Newton platform supports faxing, printing, beaming (infrared messaging), packet radio, cellular



Newton offers a rich user experience, but to achieve that requires less work from the developer. This diagram shows the relative effort that's required of the developer and of the user, with each platform. When personal computers arrived on the scene, a lot of developer effort was required to create an application—and once built, those applications still demanded a high level of savvy on the part of their users, who had to memorize arcane commands and master a complex set of operations to run an application. Operating systems were basic; virtually all the functionality of a computer came from applications themselves. Macintosh was a serious improvement for end-users, but still a significant challenge for developers—even though the operating system did much more of the work than the first PCs. Now Newton takes developer productivity to the next level: Much has been done for the developer so that applications can be developed more rapidly, and the user experience is even more intuitive, requiring little or no training.

digital packet data (CDPD), LAN-based e-mail, and commercial public e-mail. What's particularly important from a developer standpoint is that it enables you to create transport-independent applications—applications that don't have to be modified every time you want to change your means of communication. Say, for instance, you're creating a salesforce automation application that sends data back to corporate headquarters using CDPD. If, at a later date, sales decides they'd rather send that data back via a different transport, it's a simple matter to do that—without having to recode your application. The Newton Communications Architecture features an “endpoint communications interface” that makes it possible for developers to add new communications capabilities to an application or Newton PDA device. The endpoint interface provides methods for managing communications connections and sending and receiving data via modems, serial cables, and ADSP. Using the Newton endpoint interface, you can enable your applications to take advantage of technologies such as wireless modems, and alternative input devices such as bar-code readers.

- **Drivers.** The Newton operating system also includes drivers for printers, and PC Cards, so you don't need to know the specifics of the device you're using—it just works.

### System services

The Newton system software contains hundreds of routines that are organized into functional groups of services. Your application can use these routines to accomplish specific tasks:

- **View system.** Everything the user sees on a Newton PDA is called a “view”—buttons, borders, titles, and data are all views. In building an application, you create view templates to define your user interface. At run time, the Newton view system creates views from those templates.
- **Persistent object store.** Unlike personal computers that store information in rigidly defined files, Newton 2.0 provides developers with a freeform, object-oriented persistent object store that underlies all applications. The Newton Object Store provides some fundamental database storage capabilities to all Newton applications—and facilitates the sharing of data between applications:
  - The Newton Object Store enables Newton to gracefully handle the capturing, organizing, viewing, and referencing of data—seamlessly. Developers can take advantage of these capabilities without having to manage the complexities of data access and recall.
  - The Newton Object Store provides a highly efficient way to store information. Data is stored once, and referenced by each application that needs it—completely eliminating the data duplication that would drive up PDA memory and storage requirements. Of course, this approach also means that you're freed from having to implement your own data-storage functionality within applications—so development time and overall application size and complexity are reduced.
  - Information, once in a Newton PDA, can be accessed by all applications within the system, so applications can take advantage of systemwide functions—for example, Find and File.
- **Find.** Newton makes it easy for users to find specific information. It can search the data collected with a specific application, or it can conduct a global search through the data

collected with all their applications. As a developer, you can easily specify that your application is part of this world that can be searched.

- **Filing.** Filing of information is also handled at the system service level. Data in your application can be filed by the user in either application-specific or systemwide file folders for later reference, and can be put on a PC card or stored internally.
- **Routing and printing.** Routing—a function that is also built in—is the ability to print, beam, fax, or send an e-mail from any part of an application. Newton applications enable users to print to personal and networked printers; to fax via standard telephone lines, cellular modem, or GSM modem; and even to send and receive e-mail. As a developer, you'll find it's easy to support these functions in your applications.
- **Stationery.** New in Newton 2.0, stationery allows developers to easily extend applications by adding new data or views. Stationery lets developers enhance and extend existing applications without having to rewrite them. This also makes it possible to achieve a tighter integration between new capabilities and existing ones in an application. For instance, Newton 2.0 includes a choice of stationery in the Notes application—Outline for creating outlines, Checklist for keeping lists, and Notes for taking notes. As a developer, you can create different types of stationery based on the needs of your users, and extend the capabilities of built-in applications.

## Application components

Much of the hard work has been done for you. The Newton platform offers prebuilt components and a framework called NewtApp that will speed your development effort.

- **Application components.** At the highest level of the system software are more than 150 components that you can use to construct your user interface. These reusable components are neat packages of commonly needed user interface objects such as buttons, lists, tables, input fields, and so forth. These components incorporate NewtonScript code—the Newton platform's programming language—that provides standard functionality and enables the components to use other system-level services. You can also override these objects and have them behave differently, if you wish.
 

These components are built into Newton ROM. When you reference one of them in an application, the code isn't copied into your application—rather, your application simply makes reference to the component in ROM. This conserves memory at run time, but it still lets your application easily override any attributes in the built-in components. You can build much of your application using these application components, and doing so will enable you to build much smaller applications than would be possible on personal computers—or, for that matter, other PDAs.
- **NewtApp.** This is a framework that offers developers the fundamental parts or pieces of an application. By providing common application functionality that goes above and beyond the functionality provided by application components, this prebuilt Newton application framework can dramatically reduce the programming required to build an application. For many types of applications, using the NewtApp framework significantly reduces development time, because it can help you automatically manage many routine programming tasks. NewtApp offers standard application functionality, including:
  - Standard user interface elements, such as folder tabs, status bars, buttons, and close-window boxes.
  - Standard data storage, retrieval, and viewing functions, such as finding, scrolling, and



Newton Toolkit provides a palette of application components you can utilize when creating Newton applications. Some examples of these components are shown below.



Standard buttons and menus are provided for the built-in filing and routing capabilities of the Newton platform.



Sliders enable users to easily choose from a range of values.

Hello World! .....

Text input lines let users enter handwritten text that can then be automatically recognized and converted to standard ASCII text.



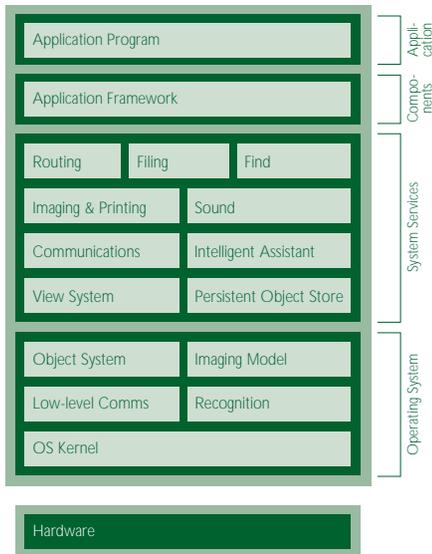
Radio buttons allow users to easily select from a number of options.



Built-in grouping capabilities enable users to organize information into categories.



Pick lists provide pop-up menus which allow users to choose from pre-defined options.



The Newton platform offers developers a rich set of built-in functionality and components.

view and data management.

— Standard communications functions, such as printing, faxing, beaming, and sending e-mail.

### Newton Tools: The right tool for the job.

The second major advantage for you as a developer on the Newton platform lies in the tools themselves. The platform provides a rich set of tools—from low-level to high-level—that let you tap into the built-in resources that the platform provides. These powerful, flexible, graphical tools can be used to quickly develop:

- **Off-the-shelf applications.** Newton tools provide the capabilities (a compiler and profiler, for example) to ensure the highest achievable levels of application performance—and to fine-tune the performance of applications.
- **In-house custom applications.** Internal corporate development teams will value the productivity benefits the platform provides—the tools enable easy, efficient creation of applications. And because of the sophisticated Newton Communications Architecture, applications can be seamlessly integrated into a wide variety of enterprise networks. What's more, since Newton applications are object-oriented, updating and modifying existing Newton applications is a simple task. Newton tools also help ensure the reliability and stability of applications.
- **Electronic forms applications.** One of the solutions areas in which PDAs are having a particularly strong impact is in the realm of forms-based applications. Solutions are being created for everything from field-sales automation to quality-control inspection and manufacturing process control. There are third-party applications for forms development.
- **Electronic books and on-line documentation.** Apple provides high-level tools for electronic book development. Newton Book Maker—a part of the Newton Toolkit—is a tool for professional publishers that enables the creation of all types of electronic books. If you don't need the sophistication of Newton Book Maker, you might consider using Newton Press—an application that offers point-and-click, drag-and-drop creation of straightforward electronic books, including graphics, tables of contents, and hyperlinking.

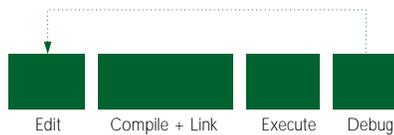
### Newton Toolkit: A powerful application-development environment.

For ISV and custom Newton application development, Newton Toolkit is the flagship tool. It's an interactive application development environment that makes it possible for programmers to design, develop, and debug applications on a Windows software-based PC or Mac OS computer, then download that application to a Newton to execute and debug.

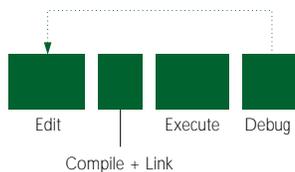
Newton Toolkit is an interactive development environment. In other words, it enables you to build your application—step by step—and test, refine, and change it as you go along. This makes it possible to create functional prototypes, test them with users, and create applications that truly meet the needs of those users. Here's what Newton Toolkit offers:

- **Graphical layout tools.** Frequently, the most time-consuming phase of application design is the creation of an application's graphical user interface. Newton Toolkit streamlines this process by providing a palette of prebuilt user interface components, as well as graphical tools that allow you to easily design the layout and user interface of your applications.
- **Reusable components.** Newton Toolkit provides complete access to the large array of prebuilt software components in the Newton ROM. Because of the Newton object-oriented architecture, you can select the components you wish to use, modify or create new components, and quickly build a functioning application. In the Newton 2.0 release, many of the platform's original components have been enhanced and optimized, and new components have been added to further streamline application development.
- **Intelligent browsers and editors.** Newton Toolkit features a variety of intelligent browsers and editors that provide a hierarchical textual representation of an application project—making it fast and easy to navigate, inspect, edit, and debug application code with a minimal amount of coding.
- **Interactive debugging.** Newton Toolkit features an interactive debugger that helps identify errors quickly. It allows you to set breakpoints and perform traces, as well as inspect and modify key values as NewtonScript code executes tasks. You can even modify your application while it is running.
- **Performance tools.** Traditionally, runtime-interpreted languages such as NewtonScript have traded off performance for ease of development and small code size. But thanks to the tools that Newton Toolkit offers, you don't have to sacrifice performance. These tools make it easier to identify and optimize the performance-critical areas of Newton applications.
  - *Profiler.* The Profiler tool integrated into Newton Toolkit makes it possible for you to quickly identify which areas of your application need performance tuning.
  - *NewtonScript compiler.* Once the performance-critical areas of an application are identified, programmers can use the NewtonScript compiler to compile those NewtonScript routines directly to machine code to optimize performance.
  - *C and C++ support.* In many cases, development teams—especially in-house development teams—have existing code written in C++. Additional Newton tools will soon be available to let you call C++ routines from within Newton applications. These tools will help you leverage existing code (and save time by not having to rewrite that code), as well as provide you with a way to run computationally intensive routines by taking advantage of C++. It is important to note that on the Newton platform C++ code is less efficient for writing and creating the user interface; C++ is best used for porting existing computationally intensive routines to the Newton platform.
- **Cross-development leverage.** Newton Toolkit is a cross-development system that runs on Windows and Mac OS computers. Using Newton Toolkit, you can leverage the large screen, memory, storage, and performance of a desktop PC to develop PDA applications. What's more, popular personal computer tools such as text editors and graphics editors can be used to create components for integration into Newton applications.

Traditional development on other platforms



Newton



The Newton platform enables developers to spend more time doing things that actually count. Compiling and Linking can be performed much more quickly—and developers can actually make changes to code on-the-fly, even while examining how an application is running.

### NewtonScript: A versatile dynamic language that makes fast development cycles possible.

An integral part of the Newton Toolkit that warrants further explanation is NewtonScript—the programming language for Newton. Though it's optimized for coding for PDAs and is tightly integrated with the Newton system itself, its syntax is similar to programming languages such as C and Pascal, and it works similarly to C++ . It is well worth mastering, because the language itself offers distinct advantages:

- **Dynamic, interactive development.** Thanks to NewtonScript, you can see the results of your changes immediately—without having to endure the compile and link cycles of traditional programming languages. The result: faster development and debugging.
- **Automatic memory management.** Some of the most common—and troublesome—bugs to resolve in PC applications are associated with memory management. Significant amounts of time and overhead are spent allocating and de-allocating memory and performing other memory-related housekeeping tasks. These tasks are critical for small handheld devices, because program, execution, and storage space often coexist on the same memory device—and therefore precious memory space must be carefully allocated. To free you from the burden of managing memory, NewtonScript provides automatic memory management. As NewtonScript code executes in a Newton system, memory is automatically allocated, de-allocated, reclaimed, and consolidated for use by other applications and system processes. The result is faster application development and debugging for the programmer—and a virtually crashproof system for the user.
- **Efficient memory use.** NewtonScript applications are stored in a compact byte-code format that is interpreted at run time. As a result, NewtonScript application packages require only minimal storage memory, and make for easy electronic distribution. Compared with personal computer applications, Newton applications are relatively small—an extensive Newton application would be only 100K.
- **Optional compilation.** To optimize performance-critical portions of NewtonScript applications, selected parts of a NewtonScript application can be compiled to machine code to enhance overall performance.
- **Memory overhead.** The Newton prototype-based object system means that instances of ROM-based objects aren't copied into RAM. This greatly reduces the memory overhead of this object-oriented system.

### Book Maker and Newton Press: Electronic book and on-line documentation tools.

Increasingly, mobile applications include significant content that's important to the user: product information, reference material, procedures and step-by-step instructions, and contact information. Included as a part of Newton Toolkit, Newton Book Maker streamlines the process of creating an electronic book, giving you a way to use Newton PDAs to distribute information to the people who need it. A Newton book can be used as is—or enhanced and extended via NewtonScript to offer users significant custom functionality. Book Maker includes support for:

- Both text and graphics data types
- Navigation and hyperlinking

- Electronic bookmarks that enable the user to mark pages for quick reference
- Annotation, so users can comment or take notes on an electronic book, using digital ink
- Easy generating of tables of contents
- The ability to print, fax, beam, and e-mail a book

Apple also offers a tool for end-users. It's called Newton Press, and it provides an easy way to graphically combine text and graphics to create electronic books for Newton. Newton Press offers an intuitive point-and-click interface that makes the process easy. It requires no programming knowledge, so it's ideal for simple electronic books with straightforward links in information. It comes with built-in translators for a number of popular Mac OS and Windows applications. Newton Press is less versatile and less powerful than Book Maker—for one thing, it doesn't give you as deep a level of control over the book, nor does it let you add NewtonScript-enabled functionality to a book. However, for many uses it is ideal because of its ease of use.

### Electronic forms tools: Creating easy-to-use mobile applications.

Notebook computers are great for many tasks, but they're hard to use while standing up. PDAs, on the other hand, because of their low weight and small form factor, are perfect for mobile applications. People who work outdoors, take inventories, or care for hospital patients are just a few examples of the mobile workers whose jobs take them away from their desks—yet who must gather, process, and record vast amounts of data. More and more companies are turning to PDAs to help these mobile workers.

An increasing array of applications for handheld devices involve the use of PDA-based electronic forms in collecting and consolidating field data—or integrating workflow processes.

Built into Newton 2.0 are many of the standard building blocks for electronic forms applications, and independent software vendors offer a variety of forms tools that allow developers to build custom electronic forms applications. Using these graphical forms tools on a Windows or Mac OS computer, you can create, customize, arrange, and link forms components—and download them for use on a Newton PDA.

Once information is gathered, it can be easily transferred to PC or host systems, thereby reducing labor and eliminating the cost, time, and errors associated with transcribing data by traditional means. For more about how electronic forms are being deployed, see Section II of this paper.

### Desktop Integration Libraries: Synchronizing data between PCs and PDAs.

To enable an application to have complete access to data stored on a Newton PDA, Apple's Desktop Integration Libraries (DILs) provide complete C-language APIs for Windows and Mac OS computers. Using these APIs, you can directly synchronize data between Newton "soups" (soups are the data sets stored in Newton devices) and a Mac OS or Windows application. DILs will enable a whole new breed of desktop and PDA applications that offer seamless data-sharing and synchronization. Several ISVs have already taken advantage of DILs technology to create applications that synchronize information between Newton applications and personal computer applications. For example, NOW Software has used DILs technology to create a synchronizing feature between Now•Contact and the Newton application Names, enabling Now•Contact and Now Up•to•Date users to ensure that their Names data contains information that is up-to-date with the information in their personal computer file—and vice versa.

#### Forms tools for the Newton platform:

- FormLogic by Wright Strategies.
- Informed Designer by Shana.
- PowerForms from HealthCare Communications

## Some books about Newton Development

### Programming for the Newton: Software Development with NewtonScript

by Julie McKeehan and Neil Rhodes  
\$29.95

ISBN: 0-12-484800-1

Includes demo version of Newton Toolkit

### Wireless for the Newton: Software Development for Mobile Communications

by Julie McKeehan and Neil Rhodes  
\$34.95

ISBN: 0-12-0484801-1

Includes demo version of Newton Toolkit

### Programming for the Newton Using Windows

(available in March 1996)

by Julie McKeehan and Neil Rhodes  
\$34.95

ISBN: 0-12-484830-3

Includes demo version of Newton Toolkit  
for Windows

### BASIC for the Newton: Using NS Basic

by John Schettino and Liz O'Hara  
\$34.95

ISBN: 0-12-623955-X

Includes demo version of NS Basic

## Support, training, and development assistance.

Apple is dedicated to making Newton developers highly successful, and has created programs to help them come up to speed on the platform, become efficient Newton programmers, and develop applications quickly. Apple offers a variety of support, training, and development assistance options. To learn more about the following information, call developer support at (408) 974-4897—or send e-mail to [devsupport@applelink.apple.com](mailto:devsupport@applelink.apple.com).

- **Developer University: Getting your team up to speed—quickly.** We offer courses with hands-on coding sessions to enable your team to come up to speed on Newton programming. Within just a few days, your team will be up and running, having mastered the fundamentals of the programming environment. We also offer self-paced learning solutions on intermediate and advanced topics—which can be supplemented with on-line support from instructors.
- **Support programs.** Apple offers developers a full set of materials and support to make you successful. All programs feature developer mailings, including quarterly developer CDs, and the bimonthly *Newton Technology Journal*. The programs also provide special access to developer information on-line, discounts on training and Apple hardware, and increasing levels of support—based on which program you choose:
  - Our self-help program provides you with information and services to help you do your own development. If you need more help, you can—for an additional fee—send questions via e-mail to our developer technical support (DTS) team.
  - Our middle-level support program—our most affordable technical support option—offers a full set of self-help tools. It also lets you ask up to 10 questions via e-mail. Our developer technical support team will handle them efficiently.
  - Our most comprehensive program offers self-help tools and provides unlimited technical support via e-mail (there is no limit to the number of questions you can ask of the DTS team). For teams that need an exclusive level of support, we offer an executive-level program that even offers code and interface design reviews.

Many of our technical and support teams have been with the platform since its inception. We are dedicated solely to Newton developers. We strongly believe our success is indelibly linked to yours, and we think you'll find that our team is highly responsive to your questions.

- **Systems integrators and consultants.** Apple works with a variety of systems integrators and consultants who have significant experience in engineering, designing, coding, and implementing Newton-based solutions. Apple also sponsors a number of support programs for systems integrators and consultants—symposia, newsletters, and programs to match partners.
- **Books.** Books have been written to bring you up to speed on the Newton platform. A list of books appears on this page.
- **User groups.** Across the country there are professional Newton user groups and developer groups that can provide key insights into questions you might have about the platform. There are also on-line Q&A areas, where you can get answers to your technical questions. Start by looking at <http://www.info.apple.com/newton>.

### In summary: Reasons to make Newton your platform of choice.

There are literally dozens of reasons to choose the Newton platform. In this section, we've covered three of the most important:

- **Better built-in functionality.** The Newton platform offers developers a rich set of functionality at every level of the platform's architecture—from the operating system to system services to built-in components. The result: It's significantly easier and faster to build Newton applications, because, as a developer, you can capitalize on the work that has been done for you. We also have a group of highly talented third-party technology partners who can provide add-ons and technologies that are not part of our base solution, but that can help you in your particular application.
- **A wide range of tools.** The Newton platform offers you both low-level tools and high-level tools—in short, you can pick the tool that's appropriate for your development effort. Whether your goal is to create an off-the-shelf application or a custom corporate application, the tools will speed your work, enabling you to see progress instantly. The Newton platform's interactive development environment provides a way to create working applications and prototypes quickly—so you can get user feedback along the way, rather than wait until an entire application is finished.
- **Superior support programs.** Apple provides a wide range of resources to help you—from training programs that will quickly get you up and running on Newton development, to support programs that will help you throughout the development process. Apple can also put you in touch with systems integrators, consultants, and other Newton platform experts who can aid your development efforts.



# Section II

Application development case studies

---



## Case Studies: Newton tools in action

Following are case studies about four development teams—and the applications they developed. In each, the development process is described, so you can see how Newton tools were used. We've chosen one case study for each of the following areas:

- ISV development using Newton Toolkit
- In-house development using Newton Toolkit
- Electronic reference materials and electronic books
- Electronic forms applications

The following pages document how Newton tools enabled these teams to create exactly what they envisioned—quickly, efficiently, and flexibly.

# Pocket Quicken

## Commercial development

What do you do when you're a market leader in the desktop PC industry, and you've already brought your software to hundreds of thousands of users on both the Windows and Mac OS platforms? If you're Intuit, the creators of Quicken, you look for customers in new markets. PDAs, of course, are a natural fit for products such as Quicken—a finance and expense-tracking application. By using Pocket Quicken on a PDA, customers are able to record expenses as they happen—a check they've written, an ATM withdrawal, and so forth. The result is that it's much easier to track expenses and keep accurate records, because users can record entries at the point of transaction—rather than wait until they return to their personal computer at home or at the office. By choosing the Newton platform, Intuit was able to achieve a tight integration between desktop PC versions of Quicken and PDA-based Pocket Quicken, enabling users to shuttle financial data back and forth between the two platforms, and keeping all entries perfectly synchronized. It was clear: "People wanted to exchange data with desktop Quicken," said Michael Tyler, Development Manager, Pocket Quicken.

### **The problem: Building a commercial application quickly.**

Intuit, like other major software developers, is constantly making trade-offs and analyzing market potential for new products—the key element examined being return on investment for programming resources and time. Intuit decided to embark on a Newton version of Quicken because they saw a potential market. In addition, they were encouraged by the high productivity of Newton tools, which meant the investment time would be worthwhile. Of course, the challenge with any major development process like this is to be able to get to market quickly—and to use precious programming resources efficiently.

### **The Newton solution: Highly productive Newton tools and superior developer support.**

The Intuit programming team got up to speed quickly, tapping into resources and technical support programs at Apple to help flatten their learning curve. The team came for developer training on the platform and tools, and then embarked on designing the

user interface for the product, taking advantage of the built-in components in Newton Toolkit. “Our entire application is built in NewtonScript, and almost all of it is built with objects that the Newton OS supplies for the developer,” said Tyler. “In the building of our interface, we used all of the different objects that are available.”

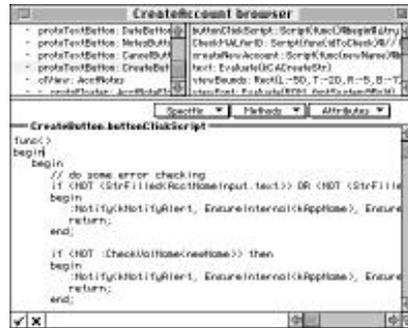
Within weeks, rather than the months it would have taken on other platforms, they were able to mock up the user interface for the product, adapting the way Quicken would function in a PDA environment. This fast interface development time enabled them to test the product with end users during the development cycle. “Given the easy and fast prototyping that the Newton environment provides, we were able to do a lot of usability prototyping—and really hone in on how users wanted to interact with the application,” Tyler said. From what they learned, the team quickly made changes to the interface, refining it at every step of the way, and tested new ideas with target customers. “During one week of testing, we were able to easily change the prototype, based on how our test users reacted to it. We were able to go in and add a button, and see if that made a difference.” The tools themselves also aided the development process. “We started working on this project in August, and we were shipping by February. It’s pretty incredible how fast we were able to develop this product—and that’s definitely due to NewtonScript and the Newton Toolkit environment,” said Tyler.

Because of the complexity and size of the development effort, it was important that the Intuit team not only have top-notch tools, but also top-notch technical support. The Intuit team felt it was important to interact with the Developer Technical Support (DTS) team at Apple. The two teams held several collaborative sessions, discussing the best ways to improve the user experience—by making changes to the interface, writing code in a specific way, and optimizing the application. The Intuit team also depended on e-mail support; questions, bugs, and problems were sent to Apple via e-mail, and a dedicated Newton support team member answered them.

The result of this collaboration is an application that runs efficiently on the platform, is easy to use, and efficiently shares data with the PC versions of Quicken. Tyler summed up the experience and overall maturity of the Newton platform: “The rest of the PDA platforms seem to be converging on the Newton development model, but Newton is certainly the most polished—and the easiest to use.”



devsupport  
@  
applelink.apple.com



**Step 1.** In deciding to develop a PDA-based version of Quicken, the Intuit developer team knew they would have to learn new tools. And they knew that to minimize the cost and time of this development effort, they would have to come up to speed on the platform—quickly. So the team headed off to Apple's training center in Cupertino, where they immersed themselves in learning the Newton development environment. During the intense training session, the team got hours of hands-on experience with Newton Toolkit and NewtonScript, the Newton programming language.

**Step 2.** The Intuit team saw the importance of getting feedback on their work from actual customers, so that customer needs would be reflected in the final design of the software. They used Newton Toolkit to get the job done. Newton Toolkit gives development teams greater flexibility during the design process. Using prebuilt components, the Intuit development team was able to quickly build prototypes of their product—and test those prototypes with users.

**Step 3.** When it came to coding the application, NewtonScript's dynamic object-oriented architecture gave the Intuit team the flexibility to shape Pocket Quicken—and to do it quickly. The team was also able to efficiently add functionality to the application, and the platform's fast compile times meant the team could move more quickly than if using more traditional tools.

**Step 4.** Throughout the design and implementation phases of the project, the Intuit team had access to Apple's Developer Technical Support team via e-mail. Intuit programmers were able to send questions, code-in-progress, and technical issues to Apple. Apple's support programs are designed in such a way that all requests are acknowledged within 24 hours.



**Step 5.** The Intuit team recognized the value of Newton-PC integration. The Newton platform offers excellent PC connectivity, so data from Newton applications can be easily transferred and synchronized with data residing in PC files. The Intuit team designed into the software a way for Pocket Quicken data to be integrated with desktop Quicken data, so that customers can take better control of their finances.

**Step 6.** Pocket Quicken was developed on a very aggressive engineering schedule. Michael Tyler, head of development for the Intuit team, said: "We started working on this product in August, and we were shipping by February. It's pretty incredible how fast we were able to develop it—and that's definitely due to NewtonScript and the Newton Toolkit environment."

**Step 7.** According to many resellers, Pocket Quicken looks to be a hit. Customers seem excited by the idea of integrating their desktop version of Quicken with one that resides on their PDA.

# Chrysler Jeep Imports, U.K.

## Custom salesforce automation

Selling is hard work. Salespeople need to master new product information, have a solid understanding of how pricing works, be able to provide quotes to customers, and take care of customers after the sale. And in no realm is there more competition than in the car market, where a number of quality brands vie for customer attention.

### **The problem: Getting accurate information to the customer—instantly.**

In the United Kingdom, Chrysler Jeep salespeople have a particular challenge, as explained by David Buckden, national service manager for Jeep. “Our success is based on person-to-person marketing into a specific niche market. Our professional salespeople need good rapport with the customer, and instant access to information on specifications and price. They don’t need a desk-based system that requires a break in the rapport in order to access this information.”

Such was the challenge in building a tool for Jeep’s sales staff: To create a highly effective and informative tool that could be used unobtrusively during a conversation with a potential buyer, or even on a test drive—without compromising professionalism. Developing that tool was not going to be easy, considering that, as John Norbury of software developer Nebu U.K. Ltd. said. “Pricing an automotive product is actually a complex process that involves negotiating a price by swapping in and swapping out different accessories to try to find a price level at which a customer will buy.” Clearly, the application would need to be flexible, intuitive, and built for a corps of salespeople whose computer skills varied. Norbury said that one of the key reasons the Newton platform was chosen was its user experience. “We deal with the assumption that 60 percent of the world doesn’t like computers and doesn’t want to use them. We design very much with the technophobe in mind. Certainly, one of the reasons we like Newton PDAs is that it provides the electronic equivalent of pen and paper.”

### **The Newton solution: A handheld information tool for salespeople.**

Chrysler Jeep worked closely with Nebu U.K. to design the Jeep QuotePad, a Newton application that gives Jeep salespeople instant access to specifications, pricing, and service information (including information on exactly what’s required for each service), and also lets the salesperson choose options for a customer and provide an accurate quote—anytime, anywhere. “The salesperson has complete information in a single source, rather than having to go to several different sources,” said Norbury.

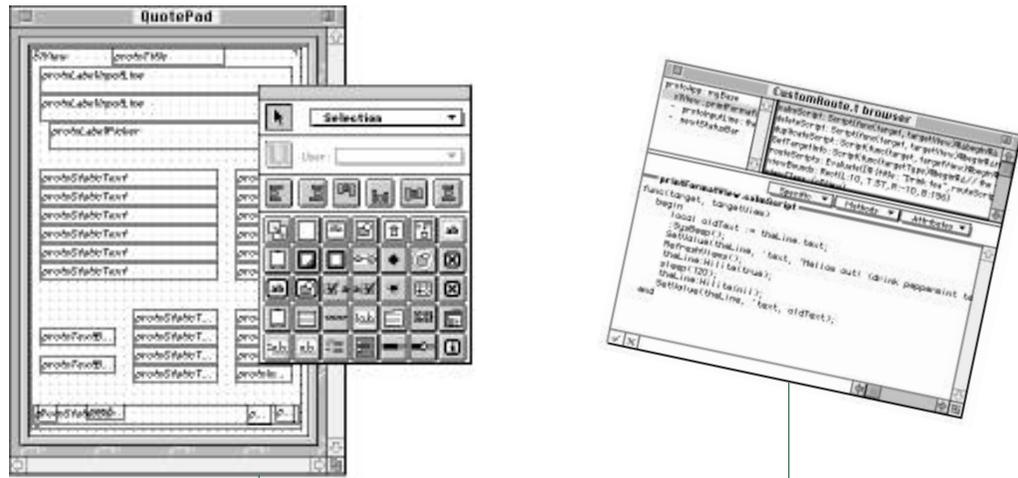
Using Newton Toolkit, Nebu was able to create the entire application—from concept to deployment at all of Chrysler Jeep's 85 dealers in the United Kingdom—in less than four months. (For a step-by-step description of the development effort, see the pages that follow.) David Buckden summed up the kind of system they were looking for: "We required a menu-based pricing system that would be simple enough to be used by the biggest technophobe."

Jeep got exactly what they were after, and it has changed the way Jeep sells. Rather than bring a customer back to a desk to crunch numbers on a desktop system, as is done at other dealerships, Jeep salespeople have the ability to work with a customer wherever that customer is most comfortable—in the dealership, on a test drive, over coffee, or wherever.

The development team analyzed how salespeople worked, and then—on their Newton PDA—provided them with smart graphics-based tools that would speed their work. Newton Toolkit enabled the creation of easy-to-use forms and reference tools. "The idea behind the QuotePad application is that the salesperson can move from one area to another by literally touching things on the screen—an invoice, for instance. That approach is fundamental to the application." With only minimal training, the sales staff was quickly up and running. "They responded very well to the system," said Norbury, "and grew to trust it very quickly."

If sales figures are any indicator, the new tools are helping: Chrysler Jeep reached sales of 4,000 Jeep units last year—twice the number they had predicted. But there has been a host of additional benefits. Because of the Newton platform's superior PC-integration capabilities, the development team was able to build a system that is also very useful for sales-team managers and Jeep's U.K. management team.

"We not only developed an application for Newton, but actually developed a PC application that links with Newton. We use the information replication capabilities that Newton provides to share information with our clients' desktop PCs," said Norbury. Indeed, the system has proven useful for Jeep's management team. In essence, as salespeople use the system, it gathers information that can help Jeep's managers. Frequent uploads of each salesperson's Newton device to the company's central computer system reveal sales patterns—whether a salesperson is assembling the right mix of options for customers, for instance. By examining these patterns, managers can coach salespeople on sales techniques and other ways they can get customers interested in Jeep's offering.



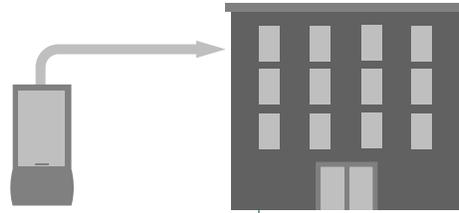
**Step 1.** As with all good applications-development efforts, Nebu started with the problem they were trying to solve: Jeep has a rather sophisticated following in the United Kingdom, and the company wanted to build a sales tool that was equal to the sophistication of their customers. They wanted to enable sales representatives to give information, including full quotes, no matter where a salesperson was—even on a test drive. The Newton platform was chosen because of its ease of use—and portability.

**Step 2.** The Newton Toolkit with its built-in functionality and ability to access prebuilt components stored in the Newton ROM made the task easy. For instance, the team was able to build the interface of the Newton application quickly. Nebu's approach was to take forms and formats that salespeople were used to (such as invoices) and build digital versions that worked more intelligently. Building the interface was achieved in weeks, rather than the months that would have been required on other systems.

**Step 3.** Using the Newton platform meant never having to start from scratch. So even though Nebu had quite a lot of data to include in the application, they were able to capitalize on buttons, menu bars, pull-down menus, and other prebuilt components. By simply typing in a name for each interface component and predetermined selections for the user, they were able to include Jeep's content—with very little actual coding. The application contained the equivalent of hundreds of pages of reference material on everything from pricing to options and even maintenance procedures.

**Step 4.** When it came to coding the application, the Nebu team also enjoyed some advantages that are particular to the Newton platform. Because of its flexible programming environment that handles many of the routine tasks for the programmer—and even lets you make changes to code on-the-fly—the team was able to quickly get the application up and running. (Start to finish, the whole application was built in less than four months.) The tools allowed the team to concentrate on solving their client's problem. Said Norbury: "Our client has been amazed by both the speed with which we've developed the application, and the nature of what it does."

QuotePad	
Customer:	Royce Walthrop
Selected Model:	Wrangler 4.0 Auto
◆ Specification/Accessories	
Alarm Pack - Remote Control	234.96
Bonnet Decal - Silver	61.03
Floor Mats - Front Pair	22.00
Soft Top Kit - Black	644.71
Towbar and Bull (2500 Kg)	217.68
A 200 Release Kit	625.73
<b>Total Cost</b>	<b>1,806.05</b>
<b>VAT</b>	<b>316.06</b>
<b>Total Price</b>	<b>2,122.11</b>
<b>Net Balance</b>	
<input type="button" value="Specify"/> <input type="button" value="Re-include"/>	
<input type="button" value="Tech"/>	



**Step 5.** Certainly one of the ingenious aspects of NewtonScript is that it enables developers to design code in a way that can be easily changed later. The Jeep application takes full advantage of the platform's object-oriented structure. Because the application is designed in discrete objects, the system can be easily updated as needs change.

**Step 6.** The result is a very useful tool for Jeep. It was deployed in all of Jeep U.K.'s 85 sales offices and is used by the entire sales staff. It provides information instantly: With no break in the rapport with a potential customer, a sales rep can access everything from mileage figures to the cost of optional equipment, and even quote a price right on the spot.

**Step 7.** In addition to helping salespeople, Nebu was able to help Jeep's management team. By taking advantage of the Newton platform's PC-integration capabilities, Nebu perfected a way to share sales data with the home office, on virtually a day-to-day basis. Salespeople upload their daily sales activities—price quotes, completed deals, and so forth—enabling the management team to look for patterns and provide sales tips, advice, and coaching for their sales staff.

**Step 8.** Results have been impressive. Not only did Jeep's sales team catch on quickly and adopt the Newton tool as their new way of doing business, but Jeep sales soared—in fact, last year in the U.K. the company sold twice as many units as had been predicted.

# Brigham and Women's Hospital and Massachusetts General Hospital

## Electronic reference materials and electronic books

Many mobile professionals need instant access to information on the job, but few professions are as challenging in that respect as medicine, in which the store of knowledge required to effectively diagnose and treat patients is overwhelming. The problem in a hospital, of course, is that key reference information is rarely available on a patient's floor, since the majority of material is tucked away in offices and libraries on other floors. "The challenge for medical professionals is overwhelming," said Dr. Steven Labkoff of the Decision Systems Group at Brigham and Women's Hospital. "There is a vast amount of information that needs to be mastered, referenced, and accessed. Medical house staff are supposed to make diagnostic decisions by being thoughtful with respect to reference information. But that doesn't always happen. Part of the reason they don't always look things up is that it's difficult to get that information in a handy form."

### The problem: Getting instant access to reference materials.

At the teaching hospitals for Harvard University's medical school—The Brigham and Women's Hospital and Massachusetts General Hospital—they have taken the challenge head-on. Under Dr. Labkoff's direction, Sandeep Shah's team at a company called K2 Consultants has developed a critically important custom Newton application for residents practicing in internal medicine.

### The Newton solution: Book Maker.

An important part of a resident's life revolves around looking up information in medical texts to properly diagnose and care for patients. Using the high-level electronic publishing tool, the team at K2 Consultants was able to create a highly effective reference tool that was deployed on Newton PDAs.

Newton Book Maker enabled the development team to efficiently convert medical and hospital reference texts into a digital format—complete with diagrams and illustrations, as well as hundreds of pages of reference material. The Newton application, built under a PDA pilot program named the Constellation Project, offers residents instant access to the texts that are most critical for them while they're on the hospital floor: The American College of Physicians (ACP) Medical Knowledge Self-Assessment

Program IX Edition, the ACP Journal Club 1993, the Electronic Monthly Prescribing Guide, the ICU/CCU Drug Reference Book, and even the Brigham and Women's Handbook and Phonebook. The Constellation Project is designed to help residents learn more quickly—and, ultimately, to provide better care for patients.

The Constellation Project was a trial to see how PDAs would work in a hospital setting. As the trial went on, Dr. Labkoff found additional functionality to include. And because of the platform's object-oriented tools, adding that functionality was easy. "We were able to take the initial Book Maker prototype and enhance and extend its functionality. We added customized data views, sophisticated indexing and searching, and even whole new functions—simply by writing new NewtonScript code," Labkoff said. "So what started out as an electronic reference tool has grown to be much broader than that. For instance, it also provides an integrated medical calculator—created with NewtonScript—that enables residents to perform calculations and functional determinations from standard clinical measurements." The point, of course, is to make this an essential tool—one that gives physicians and interns what they need in terms of resources and tools when they're on the hospital floor. "The whole premise for creating this application was that you could take this information with you and use it wherever you needed it," said Dr. Labkoff.

The platform provided some unexpected benefits to busy interns. "They ended up using the native applications much more than I had initially hypothesized," said Dr. Labkoff, referring to the Newton PDA's built-in contacts file, calendar, and electronic notepad. Assessing the overall effectiveness of the program, Dr. Labkoff said: "The majority of hospitals are technology-poor. With a solid implementation of a PDA strategy, you can get a lot more bang for your buck in terms of technology investment."

To learn how best to improve the application, the K2 Consultants team created an auditing tool that monitors residents' use of the device—so data can be gathered about which parts are most useful. Phase II of the Constellation Project will explore increasing the number of references available to physicians—not only by giving them access to information stored locally on their PDA, but also by letting them use a wireless connectivity solution to access on-line medical information directly from medical information systems. "I envision the medical profession having the ability to update chapters and modules wirelessly to keep the reference information up to date. We could even send a query to a real-time medical database and it would return a list of articles," said Dr. Labkoff. He intends to take the Constellation Project to its next phase in the coming months.



## HYPERLINKING

**Step 1.** The field of medicine requires mastery of a vast amount of information. interns and residents need to refer to dozens of medical texts in order to make accurate diagnoses. The challenge, of course, is that those reference materials are rarely archived on patient floors—so accessing reference tools when they're needed most can be difficult and time-consuming.

**Step 2.** K2 Consultants—a development team hired by Dr. Labkoff of the Decisions System Group at Brigham and Women's Hospital—developed a tool for residents. The goal: To create an intelligent tool that would provide interns virtually all the reference information the residents would need for diagnosis.

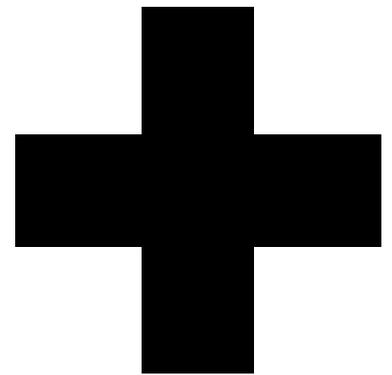
**Step 3.** The team started by taking electronic files of key medical reference books—both text and diagrams—and used Apple's Book Maker, a high-level tool for creating on-line reference books. Book Maker is an easy-to-use tool that enables users to create these on-line books, complete with hyperlinking and navigation tools for the end user.

**Step 4.** In Book Maker, it's a simple matter to format text. Unlike other on-line publishing tools, Book Maker makes it easy to designate titles, subtitles, captions, and other text levels.



```
Calculator.i Browser
beforeScript: ScriptUnit // Copyright © 1995-1996 Apple Computer, Inc. All
textText: Text("2 + 2 = ?")
viewBounds: Rect(0, 5, 100, 80)
viewChangeScript: ScriptUnit(view)
viewClass: (if #group #New)
classTitle: "Math%_Calculator%_Browser"

Math.viewChangeScript
funcCalc: view
begin
  if (calc = 'text') then
    if EndWith(self.text, "=") then
      begin
        try
          self.func := CompileSubStr(self.text, 0, StrLen(self.text) - 1);
        onexception [ext:ex] do //This should actually also catch div 0 but oh
          begin
            local e;
            e := CurrentException();
            Print("Error compiling aFunc...");
            SysBeep();
            self.lastText := Clone(self.text);
            return;
          end;
        try
          SetMenu(self, "text", NumberStr(CompileSubStr(self.func)));
        onexception [ext:ex] do
          begin
            local e;
            e := CurrentException();
            Print("Error calling aFunc...");
            SysBeep();
          end;
        end;
      end;
    end;
  end;
end;
```



**Step 5.** Book Maker also offers powerful searching capabilities—an advantage that enabled the K2 team to build an application that can be easily searched by residents and interns—so that specific information, symptoms, formulas, and diagrams can be found.

**Step 6.** The beauty of an object-oriented architecture is that as improvements need to be made, a straightforward effort is all that's required to add new functionality to an application. By using NewtonScript, K2 was able to create add-on features for the application that took it beyond being a reference tool. For instance, they added a whole medical calculator module that makes it possible to run complex calculations.

**Step 7.** The application has inspired much enthusiasm among interns and residents. Easy-to-use navigation tools mean that virtually no training is required to master this new tool; the medical staff can get up and running immediately.

**Step 8.** Dr. Labkoff has plans for further improving the Constellation Project. He is examining ways to add wireless access—so the staff would be able to view information and query medical databases throughout the hospital and the rest of the world.

# Audits International and Taco Bell

## Electronic forms applications

Many mobile professionals need to gather information on-the-fly—and then use that information to make decisions. Often, the challenge lies in how best to capture data so that it can be accurately analyzed and reported. In many cases, professionals in the field must resort to clipboard and paper to record data. But the problem with those tools is that they can introduce error into the process: In order for any data recorded in the field to be analyzed, information must first be transcribed and entered into a computer or database system. And with problems ranging from poorly recorded data to bad handwriting to the high cost of transcription, it quickly becomes apparent that paper-based systems can't meet the demands of today's fast-paced business world.

### **The problem: Ensuring accuracy and quick turnaround.**

Nowhere is performance more critical than in quality and process control. The stakes are high. Audits International, a retail quality research firm based in Highland Park, Illinois, knows that errors in the field can be costly. This was certainly clear for one of their largest clients—Taco Bell, a quick-service restaurant that serves Mexican food. Audits' 120-person quality-control team is responsible for inspecting the company's chain of restaurants. An inspector visits a chain and records data on more than 300 key variables—everything from food temperature to whether employees wash their hands frequently enough. The purpose, of course, is to get an overall assessment of a particular restaurant's health, safety, and quality standards. In the past, those inspectors used a clipboard and paper to record data; then they had to make calculations based on the data gathered, and tally an overall score for the restaurant. Next, those forms would be sent to a central data-processing center, where they would be transcribed and turned into a report for the manager of the inspected restaurant.

The process was riddled with shortcomings. At least 60 to 70 percent of the forms that were sent to be processed contained errors or lacked critical data. And resolving discrepancies—even simple addition errors—dragged out the reporting process. Often it took up to two weeks before a restaurant manager would receive a report about the inspection—and by then, conditions in the restaurant had changed, so the data had little relevance. And the volume of paper generated was overwhelming: “Each inspection required a six-page report in triplicate. With thousands of inspections performed each year, you're talking about mountains of paper,” said Gary P. DuBois, Director, Quality & Food Safety, Taco Bell Corp.

### The Newton solution: FormLogic and a smart quality control tool.

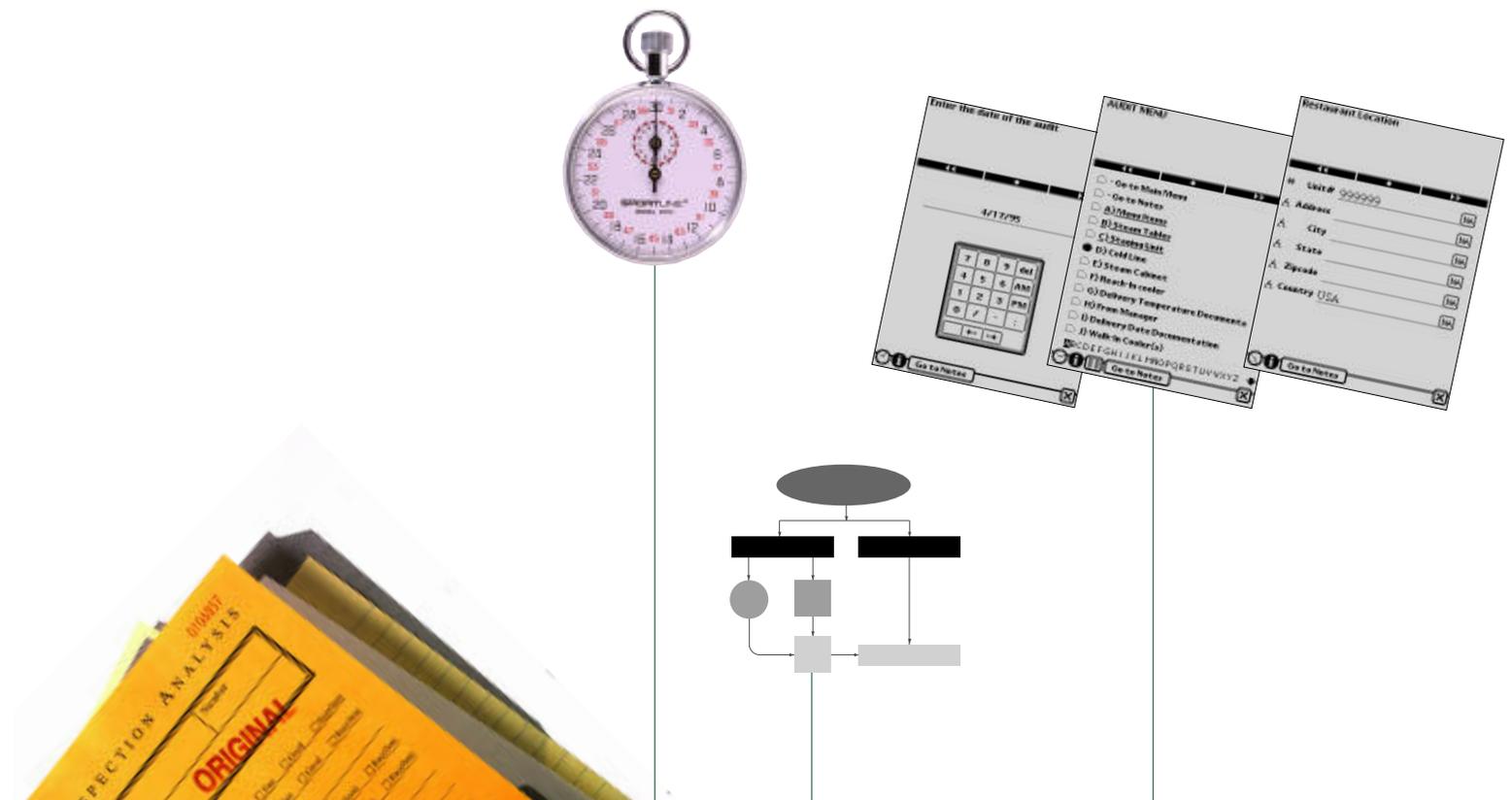
Audits International examined every part of its process—how auditors performed checks, the order in which they did the work, the types of data they gathered. The purpose was to thoroughly understand the process—to see how it could be streamlined and made more efficient and accurate by the use of PDA technology. Working in conjunction with Wright Strategies—a San Diego-based Newton developer and the creator of FormLogic software—Audits International built a handheld quality-control and inspection tool on the Newton platform.

FormLogic is a high-level tool that enables the creation of intelligent forms. Such forms don't just enable users to effortlessly capture and record data; they also automatically perform calculations, consolidate information, and prompt the user. Together, Audits International and Wright Strategies developed an easy-to-use application designed to speed the inspection process and—equally important—guarantee 100 percent reliability in the data collected.

The new system was easily learned by the auditors (in fact, most learned how to use the new Newton food-inspection system by watching a simple videotape). "The Newton platform's ease of use, portability, flexibility, and low cost made it the ideal computing platform for our mobile computing solution," said Dr. Richard Daniels, Founder and President of Audits International.

The results have been impressive. Auditors are able to conduct inspections with a high level of accuracy, thanks to pop-up menus, intelligent entry fields, multiple-choice radio buttons, and a built-in number-entry keypad. "Once the auditor records which equipment is used, the form asks questions only about the equipment that is there," said Daniels. What's more, the application performs all calculations for the auditor, and even tallies the composite score for the restaurant. A report is printed at the restaurant site, taking advantage of the Newton PDA's built-in printing capabilities—so restaurant managers have instant feedback before an auditor even leaves the premises.

At the end of the data collection, auditors use their Newton PDA and a modem to transmit data back to Taco Bell's corporate offices, where it's consolidated. "Response time is decreased, paperwork is reduced, and the results are extremely consistent," said Joaquin Pelaez, Vice President of Technology and Quality at Taco Bell Corp. "Using the paper-based system, it used to take two weeks for us to receive data on individual restaurants. Now that they're reported electronically, we typically receive restaurant results within 36 hours. And because the Newton system tallies results, we're assured that there are no clerical or mathematical errors," added DuBois.

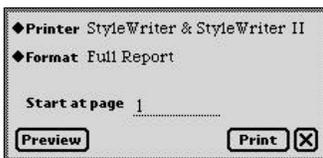
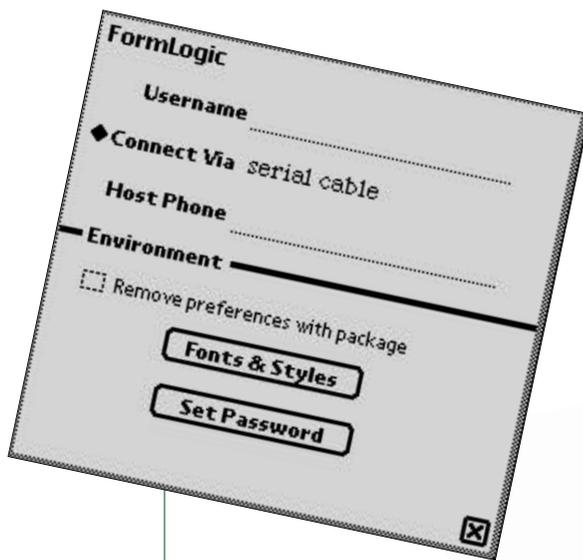


**Step 1.** Audits International decided to use Newton PDAs to improve the accuracy of their quality-control inspection process for the Taco Bell chain of restaurants. Until now, the Audits International inspectors had used a paper-based process that required inspectors to make calculations and record data by hand. The system introduced errors during the process, and turn-around was slower than desired.

**Step 2.** The Audits International PDA project had two primary goals. The first was to reduce the length of time required to complete an inspection. The second was to reduce the amount of time it took to process inspection data.

**Step 3.** Audits International hired Wright Strategies, a San Diego-based Newton developer and the creator of FormLogic. Together, Audits International and Wright Strategies examined the food-inspection process, noting which subprocesses were related to one another, and the interconnections between data gathered and the calculations that had to be made.

**Step 4.** One of the beauties of FormLogic, the tool that was used to build Audits' Newton application, is that it allows for smart agents and smart fields, completely automating calculations that inspectors need to perform.



**Step 5.** The final application is so user-friendly, in fact, that it required only a brief training video that was sent out to inspectors in the field. Results have been impressive. Inspectors easily adopted the new tool. What's equally impressive is that errors have been reduced virtually to zero. Turnaround time for processing data has also been reduced. The Audits tool takes advantage of the Newton platform's inherent printing capabilities. Inspectors can print completed forms directly from their Newton devices—in the Taco Bell restaurants. Managers at each Taco Bell have instant feedback on how their restaurant is doing.

**Step 6.** Taking advantage of the PC-integration capabilities of the platform, that same data is uploaded to central computers each night, so that reports can be generated for Taco Bell. Since no rekeying is involved, time is saved and accuracy is improved.

**Step 7.** The overall system benefits inspectors, because their work processes have been streamlined. It has improved the lives of managers because they get instant feedback on how their restaurant is doing. And, ultimately, customers benefit, thanks to a safe and clean place to eat.