



Level: Intermediate
Works with: Notes/Domino
Updated: 01-Aug-2002

by
[David](#)
[DeJean](#)

The Mobile Notes client runs on devices that shrink the resources available to a developer almost to zero, which means a one-size-fits-all approach to application development doesn't work anymore. In all fairness, a one-size-fits-all approach never completely worked. But when the varieties of the Notes client were defined by the operating system (OS) it ran on, the differences were minor—a graphics format here, a file naming convention there.

The limitations of Web browsers caused the first major shift in development strategies. The "microclient" devices that run Mobile Notes are causing a second. Currently, these devices—often referred to as palmtops or PDAs—fall into three categories, still based on the OS they run: Palm Pilots and similar devices that run the Palm OS; Pocket PC 2002, the latest version of the Microsoft OS for PDA devices; and Nokia Communicator organizers that run the Symbian OS.

Compared to the traditional Notes client platforms—PCs and laptops—these devices are defined mostly by what they don't have: they don't have big screens, for example, or full keyboards. They don't have gigabytes of storage or gigahertz of processing power either. As a result, they are limited to executing a subset of the functionality delivered by the full Notes client.

This article takes a look at the limitations these environments impose on Mobile Notes applications and gives you some tips on the best methods for developing new—or adapting existing—applications for the mobile environment. It assumes experience with designing and developing Notes/Domino applications with Domino Designer and some familiarity with PDA devices.

What Is Mobile Notes?

In spite of the limitations, Mobile Notes is a true client application that preserves the core functionalities of Notes: users can create and work with a local version of a Domino application and then synchronize, or sync, new and modified documents back to the server, where users of other Notes clients can have access to work collaboratively with the data. (The Mobile Notes documentation uses *synchronization* instead of *replicate* because the data format on the mobile device isn't an NSF file; it's XML.)

Mobile Notes is just one piece of the three components that make up Domino Everyplace Enterprise (DEE), along with sync services for Domino, and an application development tool:

- The Mobile Notes client supports a large subset of Notes functionality, so enterprise collaboration applications can be run on handheld devices.
- The development tool lets you work in Domino Designer and a Notes client to tailor your applications to the Palm, Pocket PC, and Symbian devices. It provides a framework with internal design notes, compiler documents, and filters that let a Domino server produce an XML file that contains the mobile application to be

rendered by the the Mobile Notes client. The actual data used by this mobile application is provided during the synchronization process between the Mobile Notes client and the DEE server. This is very similar to the paradigm used by the Notes client when using a local replica of a Domino application, with the major difference being the protocol used to transfer the data. It's also important to understand that the design of a mobile application and the data are separate, just as with Notes.

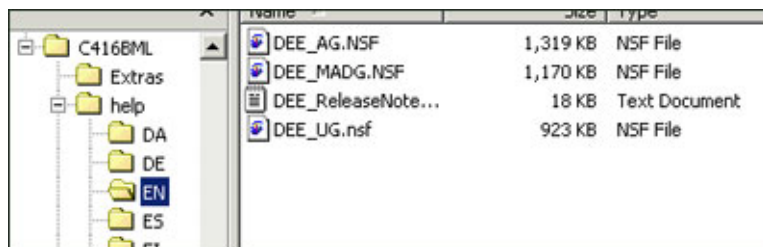
- The sync services let you easily manage both data and application transfer between Domino servers and PDAs using either wired or wireless connectivity. At its simplest, you can use the PDA's own synchronization to transfer the Mobile Notes application and data, and then use the sync services provided by DEE to update the client and server. This complex synchronization process is kicked off by a simple click of the Sync button within the Mobile Notes client. All of the underlying data and application transfer is managed by the communication between the Mobile Notes client and the DEE server.

Domino Everyplace Enterprise is installed on a Domino server running on NT 4.0 or Windows 2000. DEE requires Domino 5.0.6a or higher and the same release of Domino Designer and the Notes client. The installation creates several databases and subdirectories, including those shown here:



- Mobile Application Design is a Notes database that resides on the developer's workstation and creates the necessary documents and views in the target database.
- The DEES Administration Database is a server-based application used by administrators to manage the publication of applications and the access rights of users.
- The other three databases are documentation: a guide to the development process, a guide for administrators, and a user's guide for the Mobile Notes client.

These databases are installed on your desktop as part of the installation process. They reside, along with an essential installation Release Notes text file, in the directory structure created when you run the DEES installation. The User's Guide (DEE_UG.nsf) appears in all the various language subfolders in the Help directory. The Administrator's Guide, Mobile Application Development Guide, and Release Notes appear only in the English (EN) subfolder.



The Release Notes text file and the developer's guide database, in particular, contain information you need to know as you create and deploy applications for Mobile Notes. For more on installing and getting started with Domino Everyplace Enterprise, see the *LDD Today* article "[Creating a Mobile Notes application with Domino Enterprise Everyplace.](#)"

Mobile Notes—some Notes, some not

The Mobile Notes client running on a Palm device, a Pocket PC, or the Nokia Communicator is obviously not executing the same code used by the Domino server or the workstation clients. It is based on a small-footprint database engine that's a member of IBM's DB2 family. Also, the PDAs that Mobile Notes runs on have very different capabilities from the workstations and laptops that have been the standard target clients for Notes development. In particular, they have much smaller displays, much more limited processing power and memory, and very different connectivity models.

The result is that you'll find developing applications for the Mobile Notes environment very different, particularly if you've been developing Web applications. While most of the basic form and view functionality of Notes is available, not all @Functions and @Commands run in Mobile Notes, and there's no LotusScript, JavaScript, or

Java. (Make sure to check the Release Notes file in the Help directory for some gotchas.) More than anything else, working with Mobile Notes is like going back in time, before the Web, before home pages and navigators and framesets. It's back to the basics—forms and views, fields and columns, and some @functions to manipulate entered data.

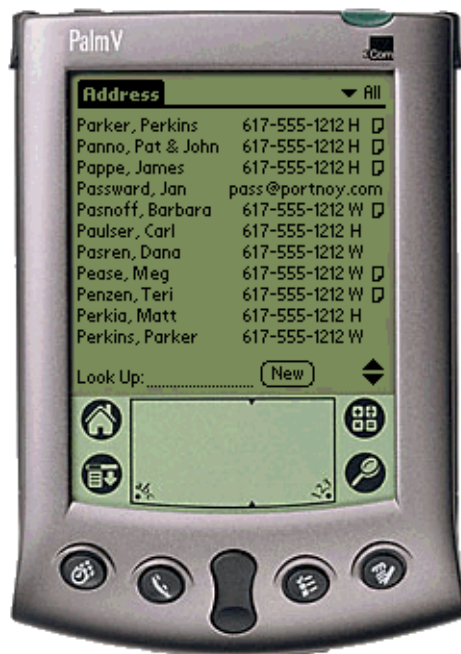
On the other hand, because you can use Domino Designer as your IDE when developing applications for Mobile Notes, you most likely already have the skills necessary to build them. It's simply a matter of building smaller, customized versions of forms and views and other design elements such that they result in a good user experience on a PDA while meeting the needs of the application.

What makes a good Mobile Notes application?

A good Mobile Notes application marries what Notes and Domino do well to what PDAs do well. These small devices typically lack almost everything that your Notes application depends on—a powerful processor and lots of memory, a big display and a keyboard—but they wouldn't be so popular if they didn't have some strengths. Above all else, they are portable. They go everywhere, so they can make your applications available in places and at times that would otherwise be impossible. They offer some ingenious solutions to data entry, such as a stylus and thumb-button keyboards. And they offer some interesting connectivity models, from wireless modems for real-time communication to desktop cradles for occasional synchronizing, with an emphasis on the easy exchange and backup of data between the PDA and a desktop PC.

The applications built into these devices maximize the value of their portability and connectivity and minimize the limitations of their displays, processors, storage, and data entry. They are models you should study closely. The calendars and address books and to-do lists and notepads have some qualities in common, qualities that you should keep in mind when you are developing a mobile application:

- They maximize the use of limited screen real estate, in part by anticipating the most common uses of the data. The Palm address book, for example, opens to a view that displays a name and contact number for each entry, and you can select a number from the several numbers in the record. The result is that you don't have to go beyond the view to get the information you most commonly want.

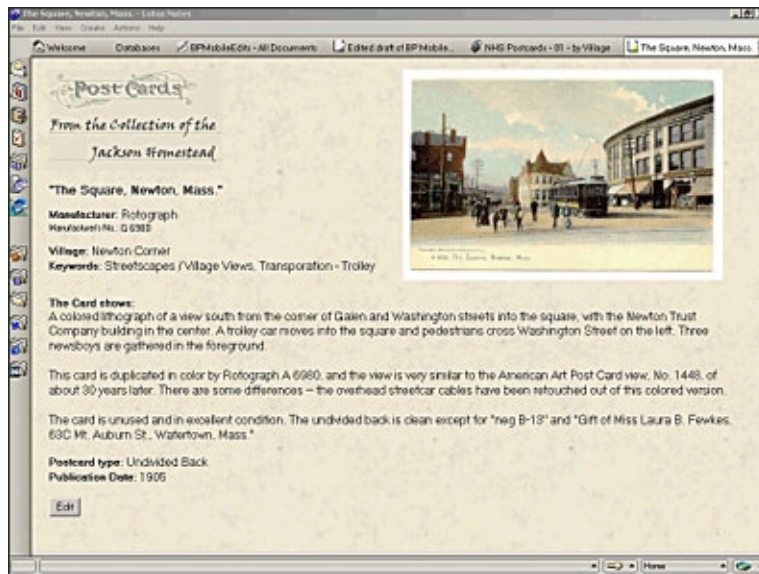


- They minimize the amount of data entry required to make them useful. The user interface for navigation to a particular record requires only the minimal number of stylus strokes that bring a name to the screen. Even creating a record in these applications requires a minimum amount of data entry to produce a useful record.
- They minimize the number of window changes within each application. The Pocket PC runs a windowing OS, but the others do not. Palm applications use dialog boxes within applications, but they can't display multiple windows. In Notes application terms, the Palm address book, for example, is essentially a set of categorized views and two forms—one for the address record and one for the attached note. The views are based on

user-determined categories:



If you're used to thinking of application development in Web terms—a frameset or a layer set that ties together navigation, indexes, and documents—you'll have to shift gears a bit to work with Mobile Notes. For example, here's a Notes application that was created for a city historical museum:



The application is essentially a specialized catalog of the museum's holdings of postcards related to the city. It was originally created to provide a convenient way to capture catalog information—subject, manufacturer, publication date, postmark date, condition, and so on—and associate it with a scanned image of the card. The form has been dressed up with a graphical heading and background image because one of the goals of the cataloging process is to make the collection available on the museum's Web site.

The application has a third purpose, as well, one that made it a candidate for being mobilized; it is a research tool. The data-entry form captures many fields that don't appear in this form, including a "further-research" field to enable collaborative question-and-answer use of the data by the catalogers. There is a large set of views, as well, so that the collection can be viewed by card type, photographer, publisher or manufacturer, series, publication, or postmark date—as well as the museum's standard set of catalog keywords.

Here's the same application running on Mobile Notes:



The 20 or so fields in the original form have been reduced to 7 on the screen (a few more are included in the form for compatibility with the catalog form). The scanned image is gone, of course, as well as the lengthy text description of the card. Information on the manufacturer and publisher has been collapsed into a single field called Maker. Also, the keyword information is eliminated. The most important field is actually something implicit in the original application—the checkbox indicating that the card is in the museum's collection. That's because the Mobile Notes application is intended to help researchers by allowing them to take the catalog into the field, to libraries, other collections, shows, and shops.

This example isn't a typical business application, but the principles are universal. You'd build a contact log in the same way: a brief form would capture the critical information about a contact with a client. It would use system defaults to enter the date and time and a series of checkboxes to indicate the nature of the contact and whether follow-up is required. When the document is synchronized with the server, it would be processed by a server agent for inclusion in a CRM application. Many other business reporting applications are similar in structure.

The limiting factor in business applications is most likely to be security. Mobile Notes applications offer the same security as Web applications; the user is authenticated against their username and Internet password from the Domino Directory for every synchronization request. In addition, each user and each device used to access the application are named in the compiler document to provide some extra security. Also, the data transferred between the device and the DEE server is encrypted.

Two key factors to consider

Two factors are particularly important to consider when creating or converting an application for Mobile Notes—available functionality and platform features.

Some Notes functionality doesn't work on a PDA. For example:

- You can't include Notes programming features that aren't available in Mobile Notes or that depend on storage or processing power that PDAs don't provide.
- You can't use some standard Web UI elements that aren't available on the microclient devices, such as framesets or complex graphics.
- You are using an off-line model. This means that there is no sense of a constant state between your mobile application and the server. So, you can't depend on a constant stream of communication with other Notes users or applications. This doesn't mean just collaboration or workflow functions are affected. Other things that require connectivity, such as @DBLookUps into other databases, are too. (However, you can use the @DBLookUp function to make calls for data either within the current mobile application or other mobile applications that are also located on the device.)

The Mobile Application Developer's Guide includes details on what Notes functionality does not work in Mobile Notes. Developers may take this functionality for granted on the desktop, but you have to rethink it when developing for a PDA. The good news is that you can create forms and views for Mobile Notes to replace or eliminate Notes functionality that's not available on PDA devices.

Less obvious, perhaps, is the idea that Mobile Notes applications should take advantage of the features of the platform. This is where careful study of other applications that run on the target device can have the greatest payoff. Lessons to learn from these applications include:

- Originality in the user interface (UI) is less desirable than consistency. Your application's user interface will be improved by your understanding of how to make it work like the other applications on the target device.
- Processing and storage requirements that might seem minimal in a desktop application will overwhelm a mobile device. Lookups and on-the-fly programming will bog down the application.
- Your application must be appropriate to the connectivity supported by the target device. Workflow applications that circulate large documents for approval and sign-off, like HR employee evaluation forms, for example, are probably not good candidates for Mobile Notes.

Best practices for mobilizing an application

This article can't pretend to be an exhaustive inventory of best practices because of the differences among the Mobile Notes client platforms, but here are some suggestions that run the gamut from global issues like application architecture to details like micromanaging screen space. The general theme is "up-leveling"—moving functionality and the delivery of information to the user up through the application architecture. Move computation from client to server; move navigation from the view to the desktop list of applications; move information delivery from the document to the view.

Use the server, not the client

Up-level computation from the client to the server wherever possible. Notes applications typically try to push computation down to the desktop for reasons of server performance. That doesn't work for Mobile Notes applications for two reasons: the Mobile Notes client may not support the functions used in the application, and there's very little processing power at the client level, anyhow. The solution is to move the processing to server agents and revamp the application design to store the results in the document rather than to calculate them on the client on the fly. The Mobile Notes filtering process can transfer only the minimum computed result necessary to the PDA, keeping the computational load on the client small and the data transfer minimal.

Remember too, that documents created in Mobile Notes and then moved to the server don't have to be the full equals of documents that have never left the desktop. The museum catalog application, for instance, keeps documents created in Mobile Notes separate and provides a manual editing process to amplify and transfer a document created on a PDA into the catalog data.

Use multiple applications to chunk the navigation

Up-leveling applies to Mobile Notes applications in a navigational sense, as well. The Notes desktop is basically a navigational tool, but many Notes users don't spend much time looking at their desktop at all. They may use a set of applications that are linked together through navigators or work in an application that performs multiple functions. A job-ticket application, for instance, may also handle the departmental work schedule, just because the ticket-assignment function depends on the schedule data.

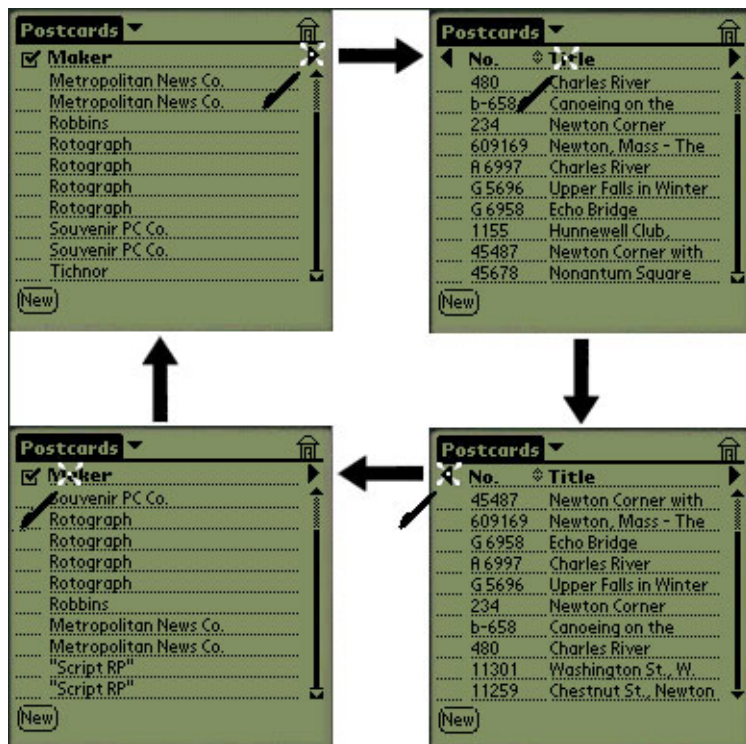
While the desktop UI may accommodate these multiple functions, the smaller screen and more limited UI of the mobile devices may not. The solution is to split these separate functions into separate Mobile Notes applications, using a feature of the Mobile Notes compiler process: A single Notes database can be used as the source for multiple Mobile Notes applications just by creating multiple compiler documents in the source application.

You can use this to create separate applications to track job-ticketing and scheduling for Mobile Notes. They will appear as two separate icons on the Mobile Notes desktop, even though the data lives in one application on the server. The division makes the Mobile Notes desktop, which users must pass through, a more useful navigation tool and simplifies the internal structure of each application, reducing the number of views and forms. The result is a simpler, more focused interface for the user.

Use the view, not the form

In Notes applications, the basic unit of the user interface tends to be the document, and views provide context and navigation. In the applications built into mobile devices, the view tends to be the basic unit, and documents provide detail. The Palm address book mentioned earlier is one example of this key difference. You should design your views with this in mind so that they can help users get to the information they seek quickly, without opening and closing documents.

On the Palm, views can be scrolled horizontally as well as vertically by tapping an arrow, and if you have enabled each column's "click on column header to sort" property, tapping on the column header will resort the column. For example, in the museum catalog sample application, you can scroll horizontally to look at the postcard number and title columns, sort by postcard title, and scroll back left to view the maker of a particular postcard. You can then return to the original order of the view, by sorting on the Maker column.



This makes the screen, even though it is small, a very mobile window on a much larger virtual view.

Use multiple views to chunk the data

Yet another up-leveling practice applies to views. Because Mobile Notes on the Palm doesn't support categorized views, views can be deep and hard to navigate. The Palm doesn't support navigators either, but it does offer a drop-down view selection menu that you can use like a navigator. You can create a set of views, each one with a selection formula that matches a category in the data. When the user clicks on the view menu arrow, the drop-down menu will work like the categorized view in the source application:



You can only use this once in an application, or you risk cluttering the view menu, but for PDA applications, once should be enough.

Use multiple forms per document

The biggest UI difference between desktop and PDA applications is in the handling of forms. The non-windowing operating systems don't scroll the screen. On the Palm OS devices, this means that a form is 160 by 160 pixels and no more. You have 13 lines of standard text, 11 of them usable by the application. (This no-scrolling limitation doesn't apply to the Pocket PC.) One way around this limitation is to create multiple forms to display different areas of the document and add navigation buttons that use an @Command, ViewSwitchForm, to move the screen window. This is similar to the ability of the Notes client to display the contents of a document through different forms.

Use spaces or tabs, not tables

The limits on form size and the length of displayed fields make views a more effective way to display data than forms. If you can, you should avoid making users open documents to find the data they want. But if users are creating new documents, they will have to do it in a form. Many of the familiar tricks for laying out forms are limited in Mobile Notes. You can't vary text color or size, or even use boldface to set off field labels, for example. (However, the Native OS field type can be leveraged to manipulate the length and height of fields in the PDA interface.) The table is one of the basic tools of form and page design for Notes and Web applications, but in Mobile Notes, the table takes up valuable screen space. Putting the seven lines of the postcard-catalog form into a table, for example, makes it too deep to fit on the screen.

Instead of tables, use spaces or tabs to align form elements. This isn't perfect because the PDAs use variable-width fonts, but it consumes the least possible screen real estate, and that will become the major criterion in your design decisions.

Summary of basic principles

Developing successful Mobile Notes applications requires careful thinking about your applications and how to match the way they work to the way the PDAs work. This article has tried to point out some basic principles:

- Understand the requirements for your mobile application. Traditional Notes and Domino applications contain a variety of application functionality, but the mobile worker may only need a few aspects of the broader application. Don't try to build everything into your mobile applications.
- Study the built-in applications on your target device. The things that make them work well will make your applications work well, too.
- The portability, small screens, and limited connectivity of the PDAs make them most suitable for reference applications that deliver small but valuable bits of information, rather than collaboration or workflow applications. (Wireless connectivity changes this equation somewhat, but security is likely to remain an issue.)
- Small devices mean small applications. In most cases, a Mobile Notes application should be based on a single view or a related set of views that categorize a dataset. If you have two very different views in your application, you should probably split it into two applications.
- The users should be able to find the data they need in the view. Only if they need to create or edit records should they have to look at individual documents.

These guidelines aren't meant to stress the weaknesses of the PDAs as Notes clients, but to underscore the fact that you'll have to think outside the box of your Notes development experience to produce effective Mobile Notes applications that take advantage of the small devices' portability and availability, and open new doors for your development skills.