EXHIBIT 63 PART 2

CHAPTER 5

Icons

This chapter describes how to design **icons**—those small pictographs that represent objects or actions in the Newton interface. Topics covered include:

- Designing effective icons
- Extras Drawer icons
- Icons in titles
- Icons in buttons
- Icons in pickers

Designing Effective Icons

This section presents some basic guidelines for designing effective icons. Remember that all your icon designs must work in the context of a Newton device. The theme of the Newton interface is communications. You want to build on this theme and diverge from it as little as possible. You must also consider other important facets of the Newton interface described here when you're designing icons.

Thinking Up an Icon Image

An icon is like the proverbial picture that's worth a thousand words only if it clearly identifies what it represents. Coming up with a tiny, grainy, black-and-white visual image that is even relevant, let alone unambiguous, can be difficult. Far more of us have learned basic verbal skills than basic visual skills. There are several approaches you can take to finding a visual image that identifies what it represents.

If an icon has some correlation to a physical device, such as a calculator or a clock, the icon should resemble the device. If the icon represents part of a metaphor, such as an in box as a metaphor for the place where incoming communications are kept, the icon should resemble its metaphorical counterpart. For example, the In Box icon resembles a real in box, not some other repository such as a folder or a flour canister.

If you need to design an icon for a more conceptual entity, such as an application or part of an application, you can use one of the following approaches. Try making the icon represent the function of the application. If the function is complex and hard for new users to understand, think about how you could explain the idea to someone who doesn't use a Newton device, and try to generate some images that way. Often the terms you use and the analogies you come up with to explain the concept can provide clues for visual images.

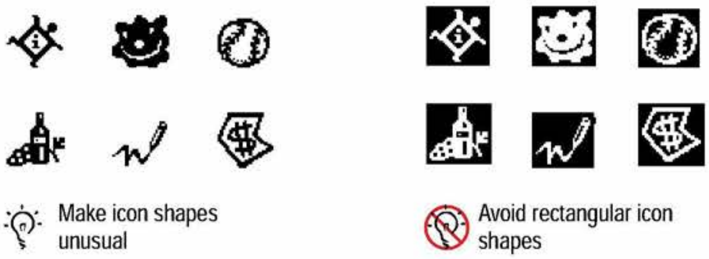
Another approach to designing conceptual icons is making the icon representative of a product name. This may work for your product in one location, but remember that some product names, and thus product icons, may be impossible to localize. For example, in the United States, an icon for extensions could have something to do with an extension cord. In other languages, the word used for extension cords may have nothing to do with extensions, and therefore an icon based on the word *extension cord* would be meaningless. Another drawback to this approach is that product names are often not finalized until late in the development process, so you might not have much time in which to design an icon based on the final product name.

It is often easiest to create icons that represent objects (nouns) rather than actions (verbs). For example, the function of deleting something is represented by a wastebasket (an object) rather than by some image of the action of deleting. Thinking of an object that is representative of the function of your icon is the key to good conceptual design. Remember that for every image you generate, you need to consider the advantages and disadvantages of the idea in regard to your audience before deciding on the final design.

Make Shapely Icons

People are good at recognizing patterns and shapes, so make the shape of an icon distinctive. Rectangular, slab-like icons all look the same, particularly without colors or shades of gray to create a pattern on them. Once users have seen a distinctively shaped icon and learned what it represents, they are likely to recognize it again and quickly recall its meaning. Figure 5-1 compares distinctively shaped icons with rectangular icons.

Figure 5-1 Distinctive icon shapes are easier to recognize than rectangular icons



Design for the Newton Display

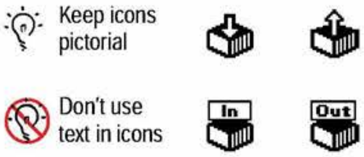
A distinctive icon shape without any detail is just a shapely shadow. When adding detail to make an icon more interesting, keep in mind the capabilities, limitations, and conventions of the Newton display. To make your icons look like they belong on a Newton, use lines that are two pixels thick. An icon drawn with single-pixel lines looks like it belongs on a desktop computer. What's more, the thicker lines are easier to see in low light. Three-dimensional effects in icons are difficult to achieve on a Newton because they require shading and many angled lines. Those effects are difficult to render on screens that display only black and white (no shades of gray or colors), particularly in the smaller icon sizes.

Newton icons do not have drop shadows. There is no assumed light source to create an artificial shadow.

Avoid Text in Icons

Avoid using text in your icons whenever possible. Text in icons can be confusing, and it's hard to localize for other regions, languages, or countries. It's appropriate to use text with icons, but not within icons. Figure 5-2 shows an example of icons with text in them and icons that convey the idea much better without text.

Figure 5-2 Avoid text in icons

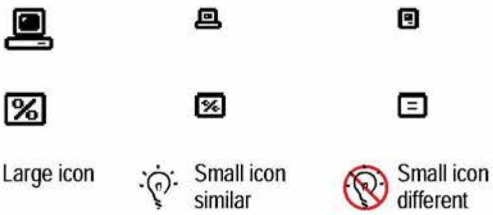


Make All Sizes of an Icon Look Alike

If you make an icon in more than one size, maintain a close visual relationship among all sizes. Design the large icon first, and then adapt the design to the small icon. You can leave out inessential details in the small version of your icon, but it shouldn't look significantly different from the large version. Figure 5-3 shows examples of small and large icons.

You can't design each variation of an icon in isolation. The large and small variants are incarnations of the same icon, so the basic design must work for all of them. Be flexible in adapting your design to all sizes. Icon design is an iterative process. During the design process, you may need to redesign one version of an icon when you find it doesn't translate well to another version.

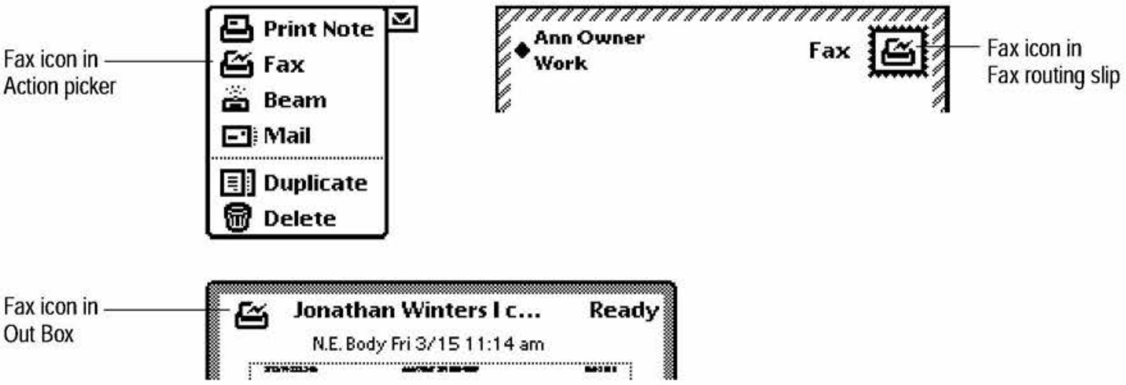
Figure 5-3 Small icon resembles large icon



Use Icons Consistently

Use icons consistently throughout your application. If there is an existing design for an icon, use it. Don't invent new designs for icons that have a standard design, such as the icons for printing, faxing, beaming, mailing, duplicating, and deleting. Figure 5-4 shows that the icon for faxing is the same in an Action picker, in the Fax routing slip, and in the Out Box.

Figure 5-4 Use icon elements consistently



Fax icon in Action picker

Fax icon in Fax routing slip

Fax icon in Out Box

People generally assume that different icons have different functions or behaviors, and they may try to find operational differences even if none exist. In contrast, where applications use standard icons consistently, users can quickly learn what those icons represent.

Think About Multicultural Compatibility

Your icons should be designed with multicultural use in mind. For example, to localize an icon for outgoing communications, you might consider using the design of a mailbox. But if you did, you would have to design a different icon for every country in which your product shipped. Instead, try to design one icon that is understood universally, or at least in many countries. An example of an icon that is understood around the world is the Out Box icon. Even though people in different locations around the world deposit mail in differently shaped boxes, they all still recognize the Out Box icon as a representation of outgoing communications.

In general, icons shouldn't be gratuitously cute. Humor typically doesn't translate well to other cultures or languages. Also, don't use inside jokes or pictures that represent code names. Although it might work to use such icons during your development process for product identification, be sure to remove them and replace them with appropriate icons before you ship your product. Symbols and colloquial language are usually culturally dependent, meaning that what one person relates to may have no meaning or may be an insult in another person's culture.

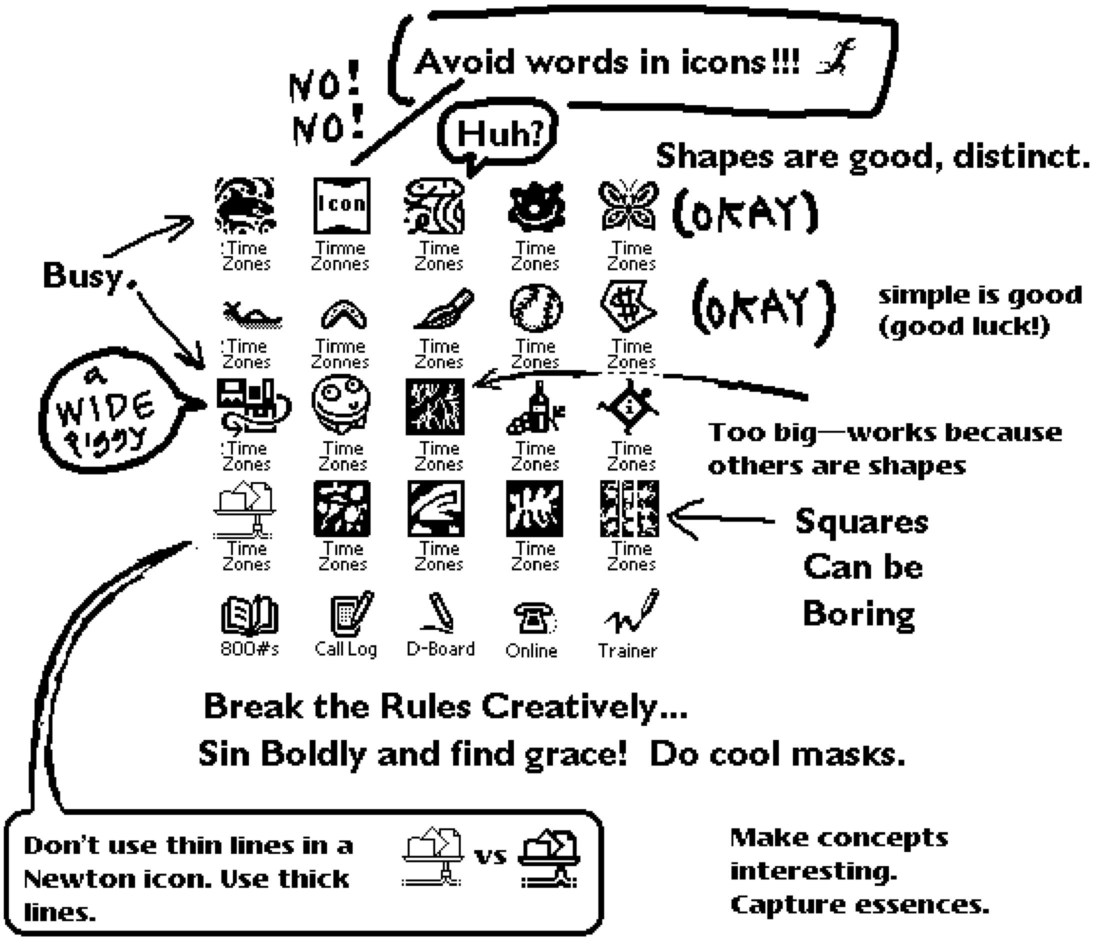
Extras Drawer Icons

For users to be able to open an application, it must have an icon in the Extras Drawer. The application name appears beneath the icon in one or two short lines of text. In addition, an application that stores data has a storage icon in the Storage folder of the Extras Drawer.

Extras Drawer Icons Together

As you design an icon for an application, look at it in the Extras Drawer next to other icons. Looking at your icon in the context of other icons may help you determine its visual impact. Is the design too light or too heavy? Is the spacing comfortable between neighboring icons? How can the icon animate to make it inviting to use? Figure 5-5 illustrates some guidelines to consider when designing icons for the Extras Drawer.

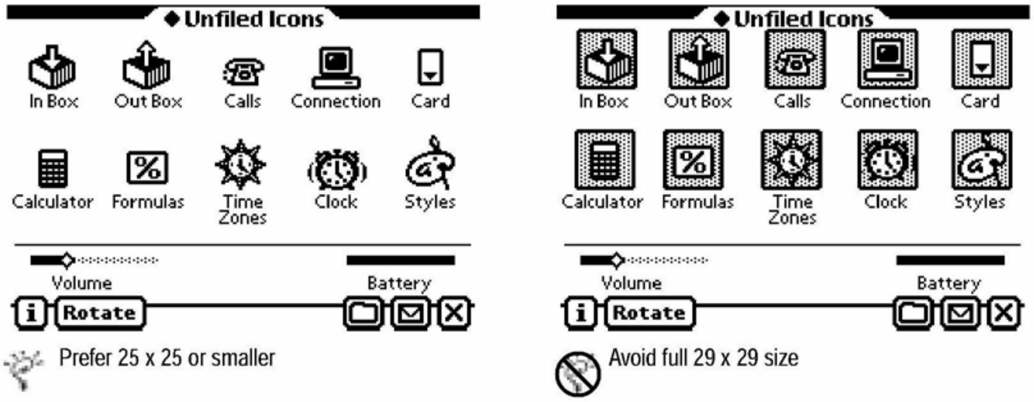
Figure 5-5 The good, the bad, and the ugly in Extras Drawer icons



Extras Drawer Icon Size

To maximize the number of icons visible at once, the Extras Drawer puts very little space between the icons in it. An application icon will be easier to recognize if it does not occupy all the space available to it in the Extras Drawer. Icons that fill their entire allotted space appear crowded and piglike. Sorry! A Newton PDA is not a desktop computer. Five icons take the same space in the Extras Drawer as just three and a half icons in a Mac OS™ folder window. Figure 5-6 compares large and small icons in the Extras Drawer.

Figure 5-6 Large icons crowd the Extras Drawer



Prefer 25 x 25 or smaller

Avoid full 29 x 29 size

On an Apple MessagePad, the maximum size for an icon in the Extras Drawer is 29 pixels by 29 pixels, but smaller icons generally look better. A full-size square icon looks too tall, and may appear to be uncomfortably close to the icons above and below it. You should give your icon breathing room by making the body of the icon no bigger than 25 × 25. Use the four-pixel margin for sparse decorative accents or for visual irregularities that extend somewhat beyond the central part of the icon and give it a distinct shape. You can also use the extra space around a 25 × 25 icon for a mask that animates the icon (see "Animating an Extras Drawer Icon" on page 5-9).

Extras Drawer Icon Shape

Icons for Newton applications generally should not look like icons for desktop computer applications. Boxy icons are common on desktop computers, where colors and shades of gray can distinguish one icon from another. In the Newton Extras Drawer, boxy black-and-white icons look too much alike, especially when they are in great number or are uniform in size. Try to give your Extras Drawer icon a distinguishing silhouette. If for some reason your design must use a black rectangular field, eliminate a pixel in each corner to make the icon more rounded. A rounded icon looks more Newtonlike.

Extras Drawer Icon Names

When you design an Extras Drawer icon, you should also come up with the name to be displayed beneath it. If the name is too long to fit on one line, the Extras Drawer automatically wraps the name onto a second line. You can control where the line breaks by including a blank space or a hyphen at a judicious spot in the name. Despite this accommodation of two-line icon names, you should avoid them. A two-line icon name crowds the icon below it and diminishes the vertical separation between icon rows.

Whether one line or two, broad icon names may collide in the Extras Drawer. To keep an icon name from running into its neighbors, make it no more than 9 to 11 characters long per line. (The length depends on which letters are in the name, since letters are different widths.)

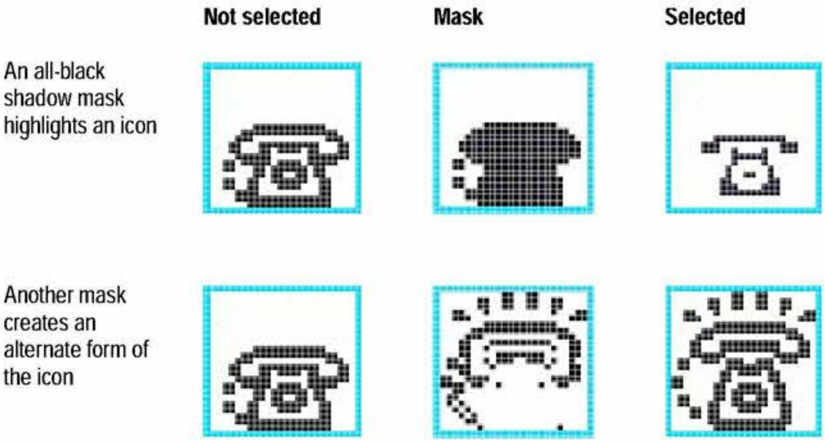
Don’t worry about your careful work devising an icon name being undone by a user changing the name. Users cannot change the names of Extras Drawer icons.

Animating an Extras Drawer Icon

Instead of having an Extras Drawer icon highlighted when a user selects it, you can have it appear to move or to change to an alternate state. For example, many of the built-in applications’ icons feature this type of simple animation, including In Box, Out Box, Calls, Time Zones, Clock, Prefs, Setup, and Writing Practice.

When a user selects an icon, the Extras Drawer creates the selected form of the icon by combining the unselected form of the icon with the icon's mask. The Extras Drawer uses the same method to animate one icon as it uses to highlight another. The design of the mask determines how the selected form of an icon looks—highlighted or animated. Figure 5-7 compares a mask used for highlighting with a mask used for animation.

Figure 5-7 An icon's mask either highlights or animates the icon

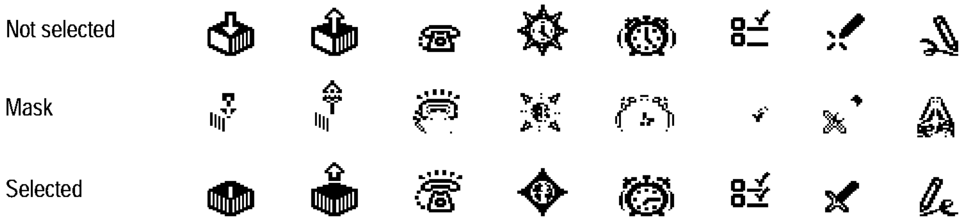

Not selected
Mask
Selected
An all-black shadow mask highlights an icon
Another mask creates an alternate form of the icon

Note: bounding boxes are for illustration only. Actual icons do not have bounding boxes.

The selected form of an icon is black only in spots where either the mask is black or the unselected form of the icon is black. The selected form is white wherever both the unselected form and the mask are black as well as where both are white.

You create a mask by comparing each pixel of the unselected form of an icon to the corresponding pixel of the selected form. If both forms of the icon have a black pixel in the same position or if both have a white pixel in the same position, the mask has a white pixel there. The mask has a black pixel where one form of the icon or the other has a black pixel, but not where both do. In making this comparison of the pixels, you are following the exclusive-or rule of logic. Figure 5-8 shows the masks of several animated icons.

Figure 5-8 Combining an icon with its mask to animate the icon

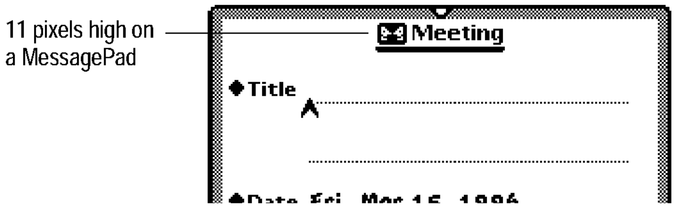


If you don't provide a mask for your application's icon, the Extras Drawer automatically creates one that is an all-black shadow of the icon. An all-black shadow mask combines with an icon to create a highlighted form of the icon.

Title Icons

An icon at the beginning of a view title graphically represents whatever the title describes in words. For example, an icon at the beginning of a slip title indicates the function of the slip. An icon is optional in a slip title. Figure 5-9 illustrates a slip title with an icon.

Figure 5-9 An icon in a slip title should decorate and inform



Meeting

◆Title

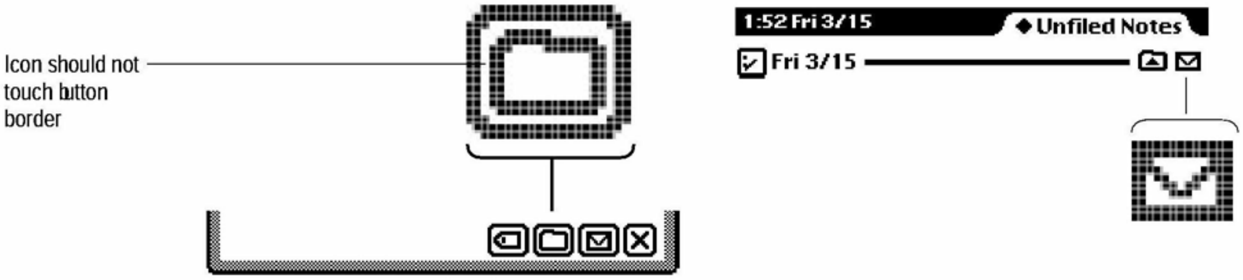
◆Date Fri Mar 15 1996

A title icon should be the same height as the title text. The usual font for the title is the 10-point bold system font, which calls for an icon 11 pixels tall on an Apple MessagePad. For more information on view titles, see "View Title" on page 2-4.

Button Icons

You can use an icon to label a button. For example, the Action button and the Filing button have icons as labels. The button may have a border or not, depending on the icon design and the button location (see "Picture Buttons" on page 3-7). Figure 5-10 shows a few buttons with icons as labels.

Figure 5-10 An icon can label a button



If a button has a border, leave white space between the icon and the interior edge of the button border. On an Apple MessagePad, leave at least one pixel of white space between the icon and the border.

Icons in a Picker

You can put an icon next to an item in a picker, although it is by no means necessary to do so. In fact, there are several reasons to avoid icons in pickers.

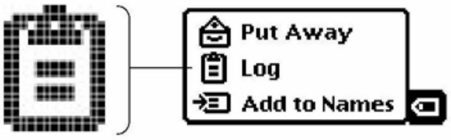
- Icons make pickers harder for new users to operate. Studies show new users initially take extra care (and time) to tap only the icons, not the text.
- Rarely can users select a picker item solely by looking at the icons and not reading the text.

- Icons increase the size of a picker, not only in width but also in height. The larger a picker, the more it obscures what's beneath it.
- If you have one icon in a picker, you have to make companions for the other picker items. It can be hard enough to state the name or function of each picker item in a word or two, let alone to design an intelligible tiny pictogram for each item.

Use judgment: long pickers may not benefit as much from icons as shorter ones. Then again, there's nothing like a good shape to grab a user's eye once the user has associated it with something the user is seeking in a long list.

If you decide to include icons in a picker, try to make them useful mnemonic devices. Use icons to clarify and distinguish the wording of picker items. Avoid purely decorative doodads. Figure 5-11 illustrates the use of icons in a picker.

Figure 5-11 Icons can help communicate picker item functions



Do not exceed 16 pixels high by 22 pixels wide; smaller is better

Icons in lists may represent individual items or they may label some attribute of each item, such as the type of item.

To be consistent with icons in existing pickers, a picker icon should not exceed 16 pixels in height or 22 in width. This does not mean you should go ahead and make all your icons 16 pixels high. Consider 16 pixels the maximum icon height in pickers, to be used only in it it's difficult to create a recognizable image with fewer pixels. Only two of the icons in the built-in MessagePad applications are 16 pixels high: the Log and Put Away items in the In/Out Box.

Where possible, vary the sizes of icons in a picker. It's not uniform size that makes icons unify a picker, give visual relief, and jog users' memories. To do that, give all the icons in a picker the same visual weight and style, but unique shapes and configurations.

The standard Newton pickers automatically align each icon with its text at their vertical midpoints. If you want to adjust the centering—visually balancing the icon as opposed to mathematically centering it—you can include white pixels at the top or bottom of the icon.

CHAPTER 6

Data Input

Although some applications for Newton devices only present information to people, many applications gather data from people as well. A person can input information in a Newton application by

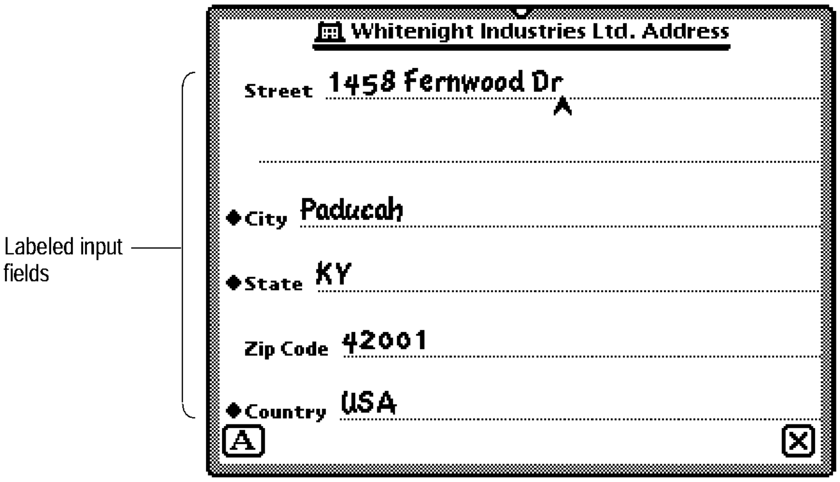
- Tapping and dragging to select an input from a list or range of options provided by the application
- Writing and drawing to input text and shapes
- Typing text on an on-screen keyboard.

The Newton interface elements that applications use for these input methods are described in the first two sections of this chapter. There's also a section on handling user errors.

Input Fields

An application gathers user input in areas called **input fields.** An input field generally consists of a text label, a value, and some means of changing an existing value or entering a new value. Figure 6-1 illustrates some typical input fields.

Figure 6-1 Users enter and edit data in input fields



Whitenight Industries Ltd. Address

Street 1458 Fernwood Dr

◆City Paducah

◆State KY

Zip Code 42001

◆Country USA

Labeled input fields

Align field labels in neat columns, and be consistent in how you align field values with field labels (including picker labels). Line up every field's label with the field's displayed value, or line up every field's label with the dotted line on which a user edits the field value.

Use the bold style of the system font for text that is the voice of the application or system, such as field labels. Use the plain style of the casual font for text that represents the voice of a user, such as values the user enters or changes. The casual font contrasts well with the system font and it has tested very well with users. On an Apple MessagePad, use the 9- or 10-point size of the system font and the 10- or 12-point size of the casual font.

Field labels are usually capitalized like sentences. That is, you capitalize the first word but not any additional words unless they are proper nouns. Do not put a colon at the end of a field label. The consistent use of fonts make it clear which text is a label and which is a value.

Tapping

People can quickly and accurately input data that an application presents in a multiple-choice format such as a picker, scrolling list, set of checkboxes, cluster of radio buttons, or slider. A user simply taps or drags to choose an input value from the options presented.

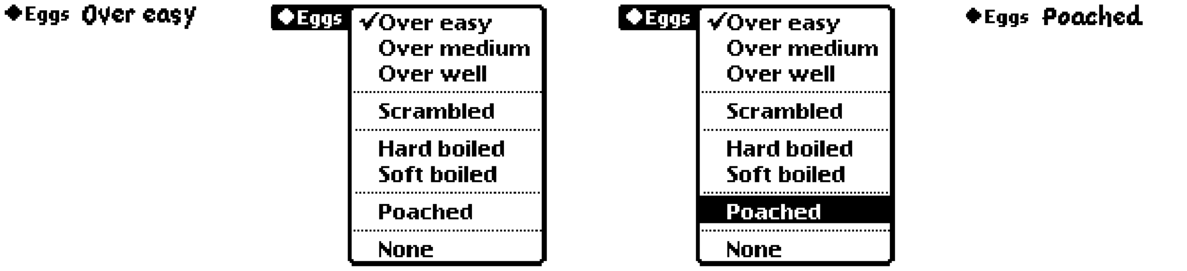
Pickers

Pickers allow a user to enter information in a way that is fast, fun, and intuitive. They have the added advantages of being easy to target and taking up minimal real estate. The Newton system defines many types of pickers: list pickers, overview pickers, location pickers, date pickers, time pickers, and number pickers. In most of these pickers, a user can input data with a couple of taps.

A picker simplifies data input by listing all the possible values, or at least several common values, for an input field. A field's picker is not visible until a user taps the field's label, which begins with a diamond. Then the picker pops up, and the user can pick a listed input by tapping it in the picker. If the user taps one of the picker items, the picked item becomes the field value and the application displays it next to the field label. If the user does not tap any of the picker items, there is no change to the selected input displayed next to the picker. Figure 6-2 demonstrates how a picker works for data input.

A picker may always list the same set of inputs, or it may list different input items each time it pops up. The application can modify a picker in response to user input or to changes in the application's environment.

Figure 6-2 How a picker works for data input



1. User taps a field label with a diamond

2. Picker pops up, and a check marks the field's current value

3. User taps a listed input to select it and close the picker

4. Selected input appears next to the field's label

For more information on pickers, see Chapter 4, "Pickers."

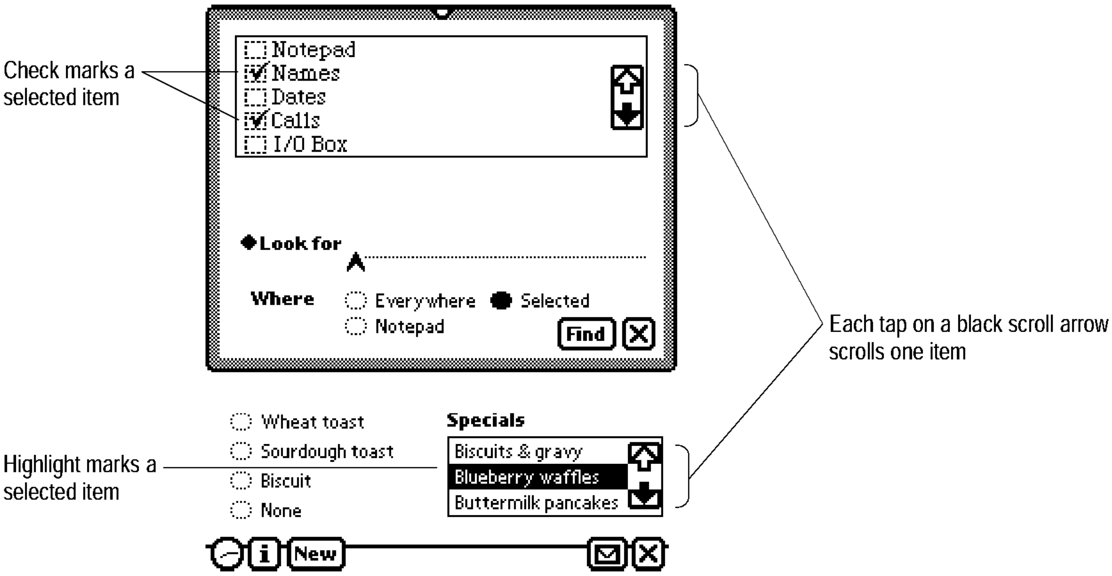
Scrolling Lists and Tables

Like a picker, a **scrolling list** is a list of items from which a user selects a field value. A scrolling list does not usually show its whole list of items at once, but a user can see items that aren't currently visible by scrolling the list with local scroll arrows. A user can also scroll by tapping and dragging the pen either above or below the list. Figure 6-3 shows examples of a scrolling list with local scroll arrows.

A user can select a listed item by tapping it, and an application may allow a user to select multiple items by tapping each item in turn. If a scrolling list includes a checkbox next to each item, then each selected item has a check mark in its checkbox. If a scrolling list does not include checkboxes, then the selected items are highlighted. Tapping a selected item deselects it. Users don't have to select anything in a list. They can just scroll through a list to peruse its contents.

A scrolling list has a thin black rectangular border with square corners. It can be any size that fits the view that contains it.

Figure 6-3 Data input using scrolling lists with or without checkboxes



□ Notepad
☑ Names
□ Dates
☑ Calls
□ I/O Box

◆Look for

Where ○ Everywhere ● Selected
○ Notepad

○ Wheat toast
○ Sourdough toast
○ Biscuit
○ None

Specials
Biscuits & gravy
Blueberry waffles
Buttermilk pancakes

Check marks a selected item

Highlight marks a selected item

Each tap on a black scroll arrow scrolls one item

If a scrolling list uses local scroll arrows, they should only appear when the list is long enough to require scrolling. Use conventional black and white scroll arrows like those used in a Find slip when the Where option is set to Selected (see “Local Scroll Arrows” on page 2-39).

Each time a scrolling list appears, it may list the same items or different items. An application can modify a scrolling list in response to user input or changes in the application’s environment.

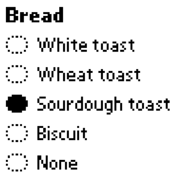
Some scrolling lists may need to accommodate items whose text is too long to fit the list. When this is the case, your application should eliminate text in the middle of a long item and insert an ellipsis (…) there, preserving the beginning and ending of the item. Text items may be the same at the beginning and middle, so if you cut off the end of the text item, users lose that context and must guess which of the several item names that begin the same is the desired one. Although an ellipsis is preferable in the middle of a long item, an application may simply truncate the item and suffix an ellipsis.

A scrolling list is not the best way to input one value across a range of values. Since the full range isn't visible all at once in a scrolling list, users have a hard time understanding the scope of their choices. Pickers work well for listing discontinuous values across a range, such as 1 minute, 5 minutes, 10 minutes, 30 minutes, and Never. Sliders work very well for displaying a continuous range of values and for letting users choose any value in the range.

Radio Buttons

For a field that can have just one of a few unchanging values, an application can use a cluster of radio buttons. A user selects an input from a cluster of radio buttons by tapping one of the radio buttons. This automatically deselects the previously selected radio button in the cluster. A cluster of radio buttons always offers the same choices; the radio buttons never change dynamically depending on context. Figure 6-4 shows a sample cluster of radio buttons.

Figure 6-4 With radio buttons, a user can select one value for a field



Bread

○ White toast
○ Wheat toast
● Sourdough toast
○ Biscuit
○ None

The selected radio button is black

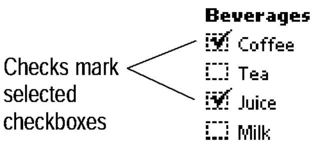
Notice that a cluster of radio buttons offers a user the same choices as a short picker. On the downside, radio buttons take up more space in a view than a picker because they are always visible in the view. On the upside, being always visible makes radio buttons faster and easier to use than a picker. There's no need to tap and a wait (however briefly) for anything to pop up.

For a detailed description of radio buttons, see "Radio buttons" on page 3-1.

Checkboxes

For a field that can have one or more of a few unchanging values, an application can use a set of checkboxes. Figure 6-5 shows a set of checkboxes.

Figure 6-5 With checkboxes, a user can select more than one value for a field

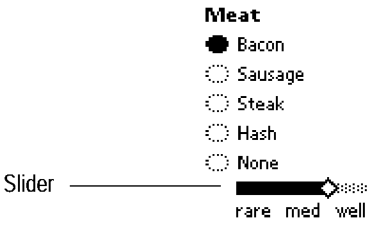


A user can select any number of checkboxes by tapping them one by one. Tapping a checkbox that is already selected deselects it. A set of checkboxes always offers the same choices; the checkboxes never change dynamically depending on context. For a detailed description of checkboxes, see "Checkboxes" on page 3-18.

Sliders

For a field that can have any value in a range, such as a magnitude, position, or probability, an application can use a slider. Figure 6-6 shows an example of a slider.

Figure 6-6 A slider used for data input



For more information on sliders, see "Sliders" on page 3-20.

Writing, Drawing, and Editing

In some places users can't be restricted to multiple-choice input methods. They must be able to input their own text or shapes (pictures). The Newton interface includes several elements in which users can write text or draw pictures. Some of these interface elements recognize text from handwriting or printing, some recognize geometric shapes from line drawings, and one interface element recognizes both types of input. All of the interface elements that recognize text or shapes also recognize a common set of gestures with which users can correct and edit text and shapes.

Text Input

There are several interface elements for text input. They can accept all types of written input, including general text, numbers, phone numbers, dates, and time. Applications can also tune the text-input elements for a specific type of text, such as numbers, phone numbers, or dates. All text-input elements have the following capabilities:

- **Recognition** Automatically transform handwriting or printing to typeset text. (An application can let users turn off or delay recognition.)
- **Ink Text** Display and store handwriting and printing exactly as written, in **electronic ink,** for later transformation to typeset text at the user's discretion.
- **Text insertion** Add newly written words at the **caret** symbol displayed on the screen or at the end of the nearest line, depending on how the user sets handwriting preferences.
- **Clipping** Automatically clip text that won't fit, by showing an ellipsis to indicate text beyond what is visible.
- **Correction** Let users correct misrecognized text.

- **Editing** Let users edit text—select, delete, copy-and-paste, duplicate, and move.
- **Formatting** Let users format individual words and characters in several different fonts, styles, and sizes.

Where there is space to write paragraphs of text (not just single lines), the text input elements also have these capabilities:

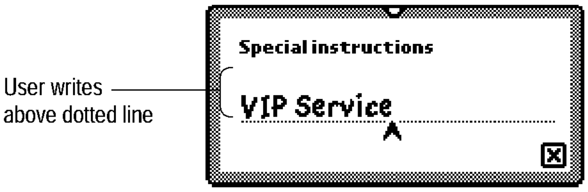
- **Word wrap** Re-form lines of text as a user writes additional words, adjusting spaces between words and wrapping words from the end of a full line to the beginning of the next line without breaking in the middle of the word.
- **Paragraph resizing** Automatically lengthen or shorten a paragraph as a user adds or deletes words from it, and allow a user to select and manually resize paragraphs.
- **Side-by-side paragraphs** Create a new paragraph alongside an existing one when a user writes words far away from the existing paragraph.

Almost all of these capabilities apply both to recognized text and to ink text. The exception: Users cannot correct ink text, since it has not been recognized. (Users can edit ink text, however.)

Simple Input Line

A simple **input line** consists of a dotted line to write on, as shown in Figure 6-7.

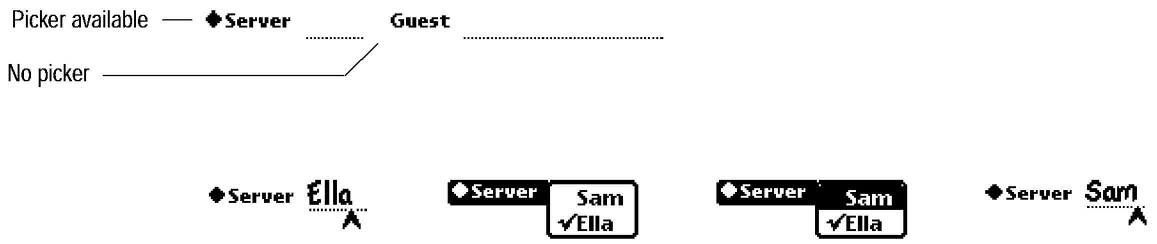
Figure 6-7 How an unlabeled text-input line works



Labeled Input Line

A labeled input line consists of a simple input line with a text label at its left. Optionally this label can have a pop-up picker that lists common values, and a user can choose one to save the effort of writing it. As usual, a diamond at the beginning of a label indicates the option of a picker. Figure 6-8 shows examples of labeled input lines with and without a picker.

Figure 6-8 How labeled text input lines work



1. User taps diamond label

2. Picker pops up, and check marks current selection

3. User taps a listed item to select it

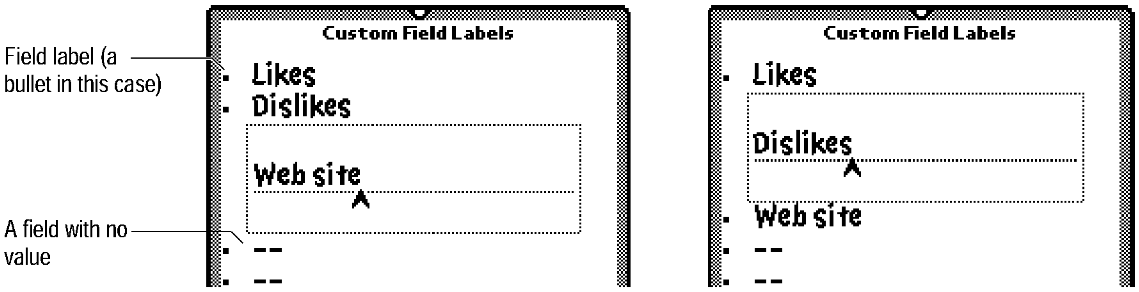
4. Selected picker item becomes the input

If you choose to implement the picker behavior, it means users can tap the label to pop up a list of input options. If a user chooses from the picker, that choice appears on the input line as if the user wrote it there. The choice is marked with a check mark in the picker. Ordinarily the chosen text replaces all text on the input line, if any. Your application can provide different behavior, if necessary. For example, choosing from the Please picker in the Assist slip inserts the chosen text at the beginning of the input line without replacing any text.

Text Input Lines that Expand

You can reduce the amount of space required for several stacked input lines in your application by using expanding input lines, which are called **expandos.** Each expando consists of a text label to the left and a text value to its right. When a user taps an expando (the label or the value), a text-input area expands from it. A user can write in the expanded text-input area, and can close it by tapping another expando. Closing an expando collapses it and updates its value. If the expando currently has no value, two hyphens are displayed. Figure 6-9 shows how an expando works.

Figure 6-9 How expandos work



1. After editing the expanded field, a user taps an input field that's not expanded

2. The previously expanded field collapses and the tapped field expands

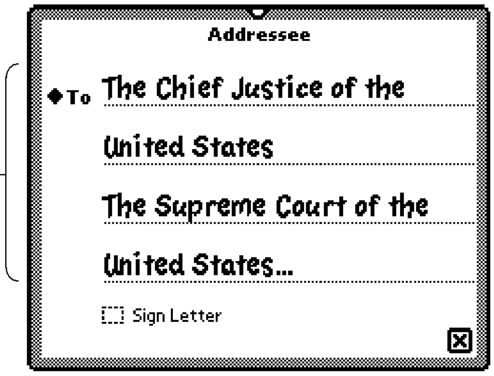
Although expandos seem to make the most of a small amount of screen space quite elegantly, they have not proven successful with users. The way they work is not particularly intuitive, and users are prone to making mistakes with expandos even after learning how to use them. Instead of expandos, you should consider using straightforward labeled input lines in slips.

Avoid including buttons or other controls in expandos. When expanded, an expando should only contain an input line. If your application needs more than that in an expando, it should be using slips instead.

Paragraph Input

Another interface element accepts the input of multiple lines or paragraphs of text. This interface element can appear simply as a blank area in which a user can write information, but usually it contains one or more horizontal dotted lines, like lined writing paper. These lines indicate to users that the area accepts input. Figure 6-10 shows an example.

Figure 6-10 Interface element for multiple-line or paragraph text input



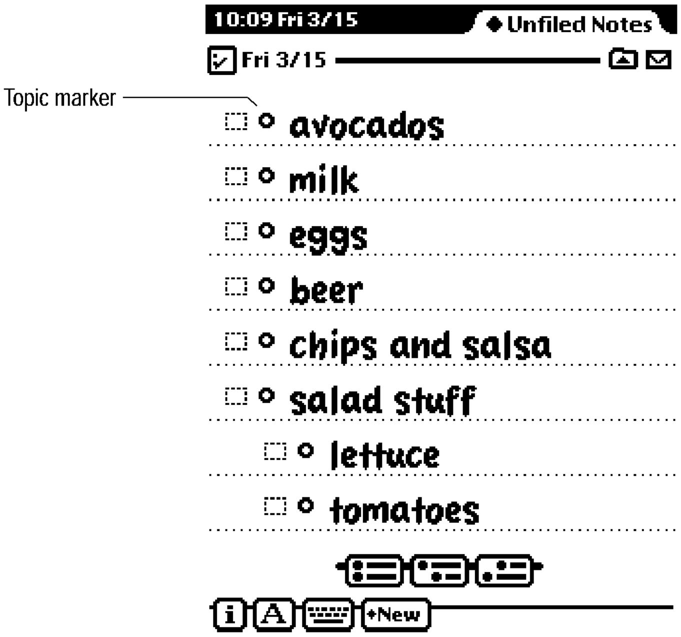
The Chief Justice of the
United States
The Supreme Court of the
United States...

User writes anywhere

Structured List Input

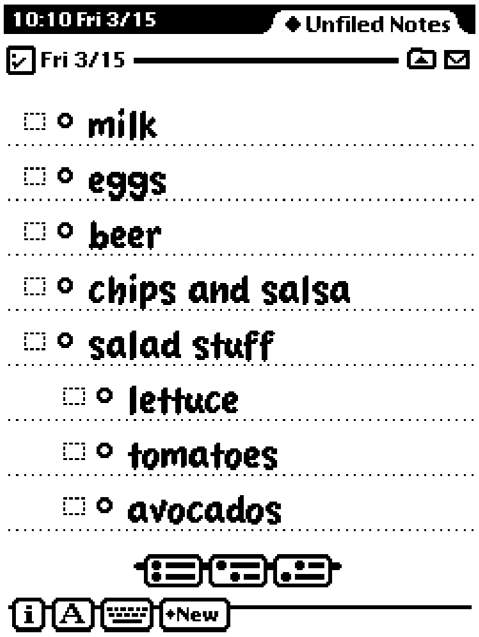
The Newton interface also includes an element for the input of structured lists such as outlines and checklists. A structured list consists of a sequence of topics, each of which may be one or more paragraphs long. To the left of each topic is a small circular topic marker and an optional checkbox. A user can drag a topic's marker to move that topic above or below another topic. A user can make a topic subordinate or not subordinate to the topic above it by dragging its marker right or left. This ability to create a topic hierarchy can be suppressed by an application in any structured list. Figure 6-11 shows an example of a structured list view.

Figure 6-11 A user can rearrange a structured list by dragging topic markers



- □ ○ avocados
- □ ○ milk
- □ ○ eggs
- □ ○ beer
- □ ○ chips and salsa
- □ ○ salad stuff
  - □ ○ lettuce
  - □ ○ tomatoes

1. Before dragging the first topic marker



- □ ○ milk
- □ ○ eggs
- □ ○ beer
- □ ○ chips and salsa
- □ ○ salad stuff
  - □ ○ lettuce
  - □ ○ tomatoes
  - □ ○ avocados

2. After dragging the first topic marker directly below the last topic marker

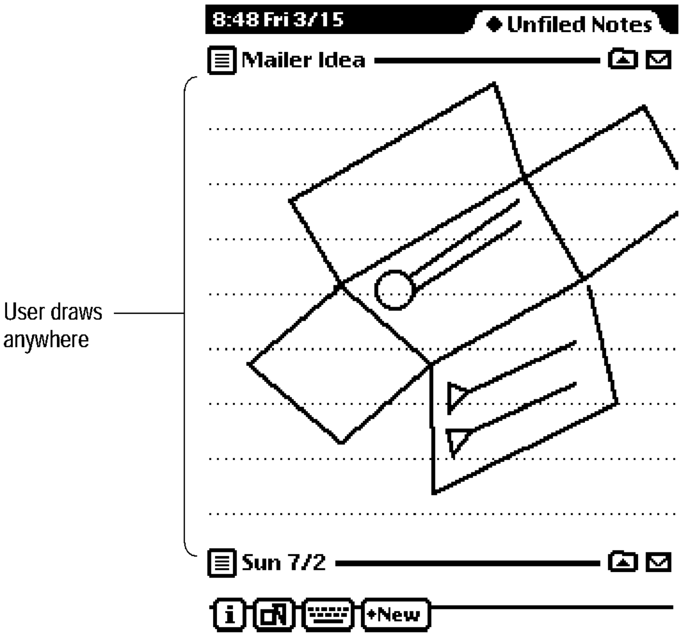
Shape Input

There is one interface element for the input of geometric shapes. It can be a blank area in which users can draw, or it can contain dotted lines to cue users that the area accepts input. Figure 6-12 shows an example.

Shape-input areas have the following capabilities:

- **Recognition** Automatically recognize geometric shapes such as rectangles and other polygons, circles and ovals, and straight lines and curves.
- **Gravity** Automatically snap new line endpoints to nearby corners and midpoints of existing shapes.

Figure 6-12 Interface element for shape input

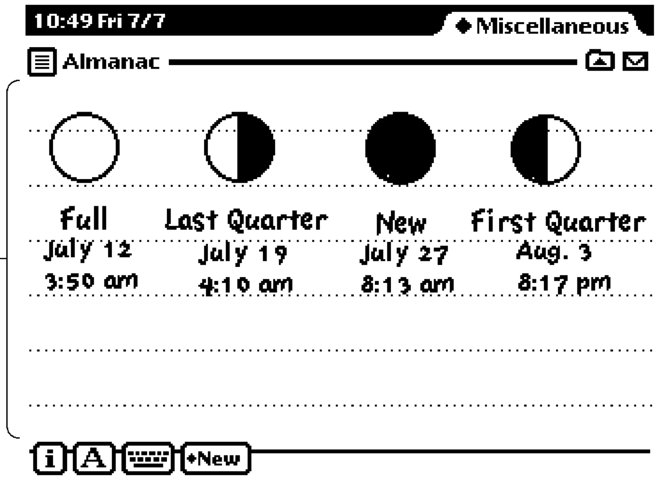


- **Editing** Let users edit shapes—select, delete, copy-and-paste, duplicate, reshape, resize, and move.
- **Formatting** Let users set the line thickness of individual shapes and shape segments.

General Input

The one interface element for general input lets a user write text and draw shapes. If the user writes text, the general input element creates a paragraph and operates within that paragraph like the paragraph-input element described in "Paragraph Input" on page 6-12. If the user draws shapes, the general input element creates a shape-input area and operates like the input element described under the heading "Shape Input" on page 6-13. Figure 6-13 shows an example of a general input element.

Figure 6-13 Interface element for general input



User draws or writes anywhere

Recognition

The Newton operating system is able to recognize handwriting, printing, and drawing, transforming it into typeset, editable text or editable geometric shapes. The Newton system can also recognize a common set of pen gestures for correcting and editing input. The recognition system has a sophisticated multiple-recognizer architecture. There are separate recognizers for words, geometric shapes, and gestures, any of which an application can make simultaneously active. An arbitrator in the recognition system examines the results from simultaneously active recognizers and returns the recognition match that has the highest confidence.

The text recognizers can handle printed, cursive, or mixed handwriting. They can work together with built-in dictionaries to choose words that accurately match what a user has written, and they can recognize a user's writing letter by letter. A user can also add new words to a personal dictionary.

The shape recognizer recognizes both simple and complex geometric objects, cleaning up rough drawings into shapes with straight lines and smooth curves. The shape recognizer also recognizes symmetry, using that property, if present, to help it recognize and display objects.

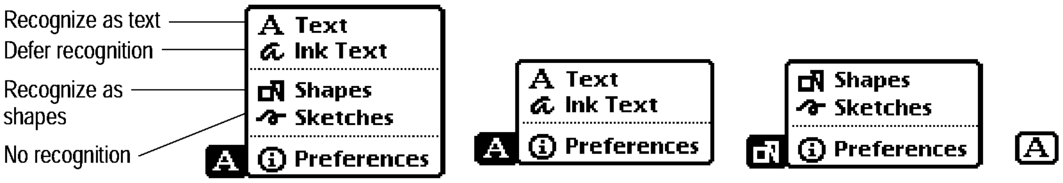
You don't need to do anything in your application to handle ordinary recognition. The Newton system's input interface elements handle recognition of writing and drawing, including a method for users to correct misrecognized words. For example, when a user writes a word on a labeled input line, that interface element automatically passes the pen strokes to the system's text recognizer, accepts the recognized word back, and displays the typeset word. If the recognizer misreads a word, the user can double-tap the word to bring up a list of other possible matches for it, or to use the on-screen keyboard or Corrector view to correct it.

User Control of Recognition

Recognition is modeless. That is, users do not need to put the system in a special mode or do all their writing and drawing in a special slip. Users can write in any text-input area and draw in any shape-input area. They can also write text and draw shapes in any general-input area.

An application that has general-input areas needs to allow users to designate whether to use the text recognizer or the shapes recognizer. In addition, users should be able to temporarily disable the text recognizer or the shape recognizer as appropriate for the type of input area. The customary way to provide control over recognition is with a Recognizer button and picker, as shown in Figure 6-14.

Figure 6-14 The Recognizer button and picker give users control over recognition



Recognizer picker for writing and drawing

Recognizer picker for writing only

Recognizer picker for drawing only

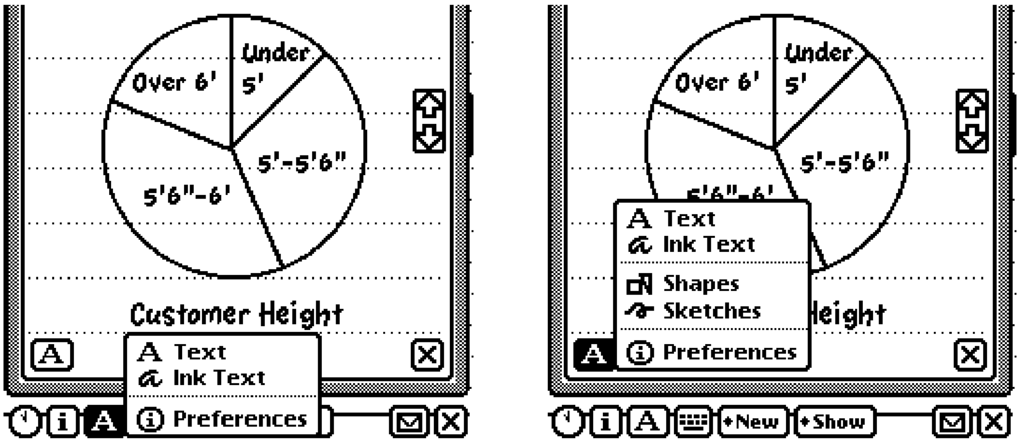
Recognizer button identifies the recognition in effect

The Recognizer picker lists the type of recognition options that are appropriate for the type of input users can make. If users can only write text, the recognizer should only include text-recognition options—Text and Ink Text. If users can only draw shapes, the recognizer should only include shapes-recognition options—Shapes and Sketches. The Recognizer picker should include all recognition options if users can write text and draw shapes.

The last item in the Recognizer picker is always Preferences. Selecting it gives users quick access to the handwriting recognition section of the built-in Preferences application.

The Recognizer picker usually pops up from a Recognizer button on the left side of the status bar (next to the Info button). In addition, users may need to control recognition separately in slips. The Recognizer picker should also pop up from a Recognizer button in the lower left corner of a slip that allows users to input text or shapes if that slip covers the status bar and can't be moved out of the way. A slip should also contain its own recognition control if it allows both text and shape input but the Recognizer picker on the status bar only offers text-recognition options. Figure 6-15 shows a shows a slip with its own recognition control.

Figure 6-15 Users may need to control recognition separately in a slip



In this application, the Recognizer picker on the status bar only controls text recognition

Another Recognizer picker controls text and shapes recognition in a slip that allows writing and drawing

Deferred Recognition

A user can defer text recognition by selecting Ink Text from a Recognizer picker. While recognition is set to Ink Text, the Newton system recognizes word boundaries but does not recognize words, letters, numbers, or symbols themselves. Later a user can double-tap ink text to have the Newton system recognize it. The user can double-tap one word at a time or can select a group of words and have them all recognized by double-tapping the selection. As the Newton system recognizes a word, it displays a small curved arrow above it and then replaces the ink text with typeset text. When a user double-taps a selection of several words, the Newton system recognizes them all, displaying a small curved arrow above each one in turn, before replacing the ink text selection with typeset text.

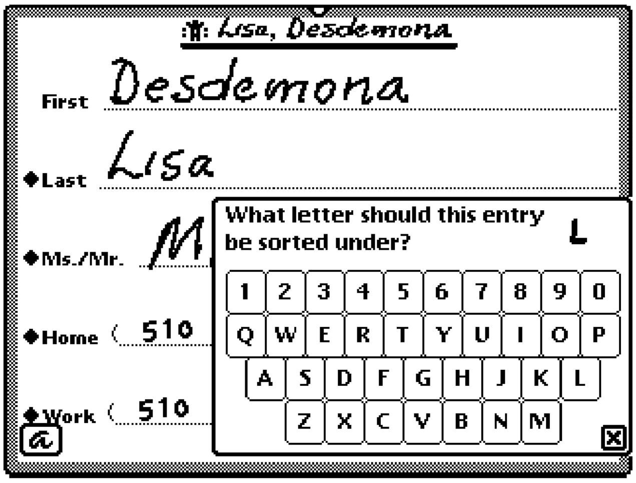
In general, ink text can appear anywhere regular text can appear, and the two can be mixed. Your application should permit ink text everywhere it accepts general text entry. There are times when users need the speed of ink text, and there are users who prefer ink text over recognized text. For those users, ink text is an important data-input tool. If your application does not allow ink text, you have needlessly limited the number of satisfied customers.

Of course there are exceptions to allowing ink text universally. An application that needs to make decisions or perform calculations based on what users write can't allow ink text. For instance, the built-in Formulas application doesn't and shouldn't allow ink text, because it makes calculations based on values that users write. Similarly, an application that keeps track of mileage might allow ink text in user comments or notes, but not elsewhere. Yet that application would be more versatile if it allowed ink text everywhere and deferred calculations based on ink-text values. A user in a hurry could jot down information in ink text without worrying about recognition errors. Later the user could double-tap the ink text to have it recognized and the application could make calculations based on the newly recognized text.

Your application does not have to disallow ink text in input fields that may be used for sorting or sequencing. If a user writes ink text in such a field, your application can display an Alpha Sorter picker, in which the user selects a sort key. Figure 6-16 shows how the built-in Names File application uses an Alpha Sorter picker if a user writes ink text in the Name field (which determines the card's sequence).

If your application simply sorts ink text with recognized text, the ink text comes before the recognized text that comes first alphabetically.

Figure 6-16 In an Alpha Sorter picker, users select a sort key for ink text



Lisa, Desdemona

First Desdemona

◆Last Lisa

◆Ms./Mr. M

◆Home (510

◆Work (510

What letter should this entry be sorted under? L

Forcing Recognition

Under some circumstances your application may have to recognize ink text forcibly. For example, when a user tries to place a call to an ink-text phone number, the Calls application forces recognition of the ink text. If your application forces recognition, make sure it allows the user to correct mistakes in the recognized text before it acts on the text.

Configuring Recognition

You can configure recognition separately for each field in your application. For example, in one field you could disable recognition of shapes and configure the text recognizer to recognize only names, and in another field you could configure recognition differently.

No matter how you have configured recognition for a text field, users can input the wrong type of text if they try hard enough. For example, a user may manage to input numbers where words are the proper type of input. This happens because every kind of text recognition uses the built-in symbols dictionary, which includes all digits and some punctuation together with all letters of the alphabet. Users have to try hard to input an improper type of text because recognition is weighted towards the proper type of text, and that is what the recognizer usually returns. But if an input stroke is vague enough, the recognizer might return an improper type of text, such as a number instead of a letter. Thus users can manage to write virtually anything in any text field.

Here is a complete list of recognition options that applications can control:

- Allow users to control recognition with a Recognizer button on the status bar, in a slip, or both.
- Separately enable or disable recognition of words, shapes, and editing gestures.
- Recognize words letter-by-letter without using dictionaries.
- Improve recognition of complex stroke groups in which users tend to put large spaces, such as telephone numbers. This is accomplished by disregarding spatial cues (distance between gestures or strokes), and relying solely on temporal cues (time between the end of one stroke and the beginning of the next one) to determine when a user has completed a group of strokes.
- Make the Names dictionary or any other built-in dictionary the primary dictionary. (The Names dictionary contains common names for a user's locale, not the names contained in the built-in Names application.)
- Use custom dictionaries with specialized words such as product names or plant species.
- Limit word recognition to one or more of the built-in dictionaries (proper names, surnames, first names, names of days and months, names of countries, names of states, names of cities, names of companies, abbreviations of states, common words, and user's words).

- Recognize punctuation marks. Preceding a word: single quotation mark, double quotation mark, left parenthesis, or hyphen. Following a word: single quotation mark, double quotation mark, right parenthesis, hyphen, period, comma, exclamation point, question mark, colon, or semicolon.
- Force capitalization of the first letter of every word.
- Recognize numbers, including monetary amounts, decimal points, and signs (+ and –). (Can't be used in combination with letter-by-letter recognition.)
- Recognize phone numbers. (Can't be used in combination with letter-by-letter recognition.)
- Recognize dates. (Can't be used in combination with letter-by-letter recognition.)
- Recognize times. (Can't be used in combination with letter-by-letter recognition.)

Editing

Users must be able to change input they have written, drawn, or typed. In all the Newton input areas described in this chapter, people can use consistent techniques for the following data-editing actions:

- Select text and shapes
- Join words
- Break one paragraph into two
- Erase text or shapes
- Insert space in text
- Insert new text
- Replace text
- Correct misrecognized text
- Change capitalization of text
- Change paragraph margins

- Remove extra space from paragraphs
- Duplicate text or shapes
- Change shapes
- Move objects

The techniques people use for these editing actions are described in the next 12 sections (ending with "Moving Objects" on page 6-32).

To make these editing actions available in your application, you don't have to do anything at all as long as you use standard input elements based on Newton prototypes. Although you can make some of the editing actions available in custom input areas, you can't make them all available. In particular, it's impossible to implement the customary techniques for selecting multiple objects, moving objects, changing paragraph margins, and more.

Selecting Text and Shapes

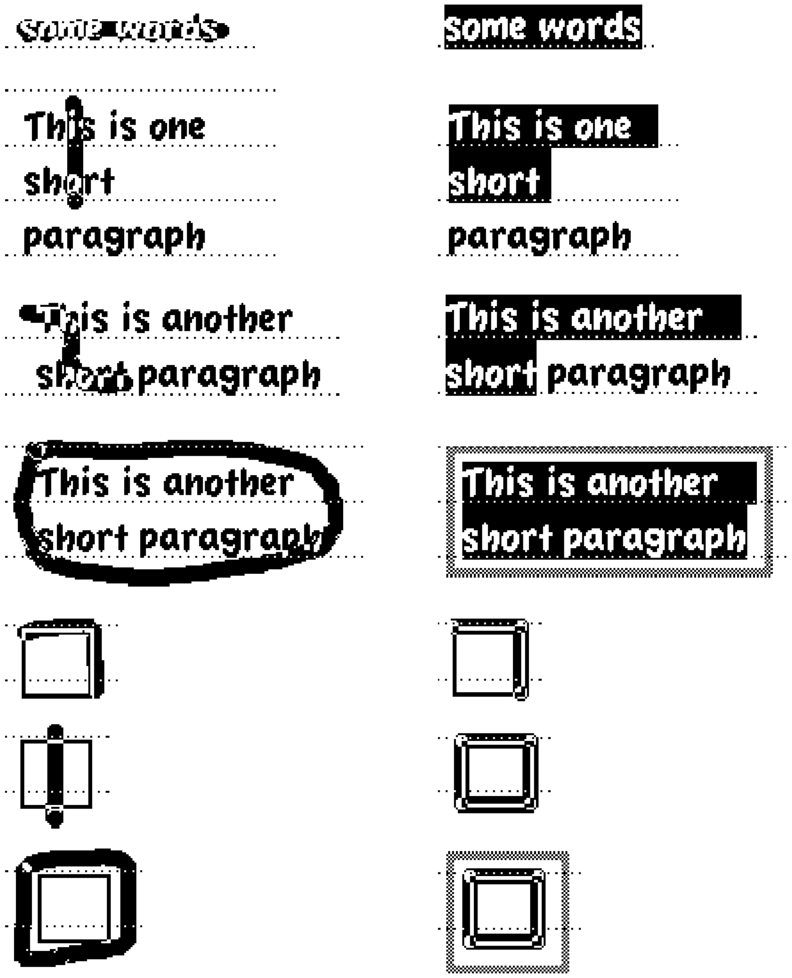
Users must select text or shapes before copying, moving, or otherwise manipulating them. To select an object, a user holds down the pen on or near the object until a heavy mark appears under the pen and the Newton device makes a high-pitched sound. Then the user draws the highlighting mark over or around the object. (The sound does not happen if the "Pen sound effects" option is turned off in the Sound section of the built-in Preferences application.) Figure 6-17 shows how selection works.

To select words, a user draws the highlighting mark across them. To select text that's on more than one line, a user draws the highlighting mark from the beginning of the first word to the end of the last word. To select several whole lines of text, a user draws the highlighting mark vertically through the lines of text.

To select lines in a shape, a user draws the highlighting mark along the lines.

To select whole paragraphs, shapes, or a combination of text and shapes, a user circles them with the highlighting mark. The highlighting mark doesn't need to be close to the items, as long as it encloses them completely and doesn't enclose anything else. The Newton system puts a gray selection box around the object or objects the user circled.

Figure 6-17 Selecting words and shapes



1. User draws highlighting mark

2. Newton highlights selection

A user can extend a selection or select more objects by drawing additional highlighting marks. If they are far apart, the user may select one at a time. Selected objects do not have to be adjacent, but all selected objects must be in the same input area. Anything that is selected remains selected when the user selects more in the same input area. If a user selects nonadjacent objects,

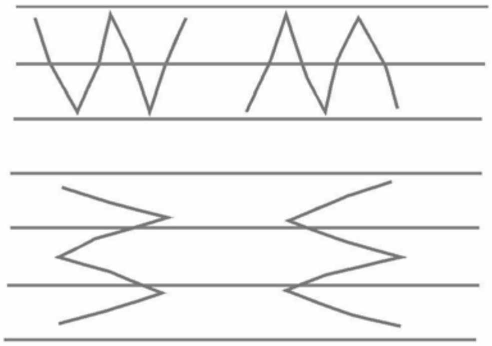
objects that the user has not selected may appear within the borders of the gray selection box, but only the selected objects are highlighted.

Erasing Text or Shapes

To erase text or shapes, a user scrubs them out with zigzag gestures. Immediately after scrubbing, the user hears a poof sound and sees smoke clouds cover the scrubbed objects. The smoke quickly dissipates to reveal the scrubbed objects have disappeared.

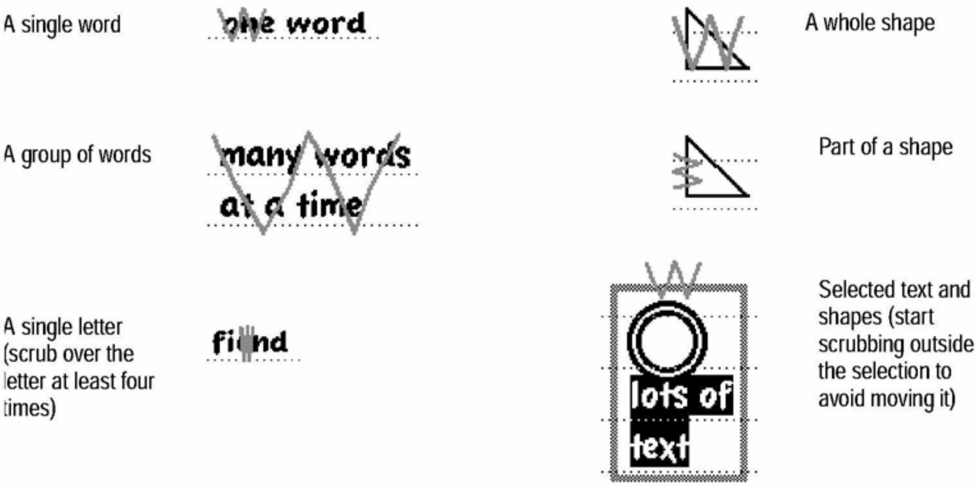
A zigzag must go back and forth at least four times to be considered a scrubbing gesture. The zigzag should have sharp corners, and each segment of the zigzag should be about the same length. The zigzag can have any of the four orientations shown in Figure 6-18.

Figure 6-18 Orientations of the scrubbing gesture



Depending on the size of the zigzag, it can erase a letter, a word, or a group of words. Also depending on its size, one zigzag can erase a whole shape or part of one. In addition, a single zigzag can erase text and shapes that have been selected. Figure 6-19 shows examples of scrubbing a little and scrubbing a lot.

Figure 6-19 Scrubbing a little or a lot


A single word
one word
A whole shape
A group of words
many words
at a time
Part of a shape
A single letter (scrub over the letter at least four times)
fiend
lots of
text
Selected text and shapes (start scrubbing outside the selection to avoid moving it)

The effect of scrubbing may be different if a user first selects several objects. If a user selects several objects, contiguous or not, and then scrubs over all or any part of the selected objects, all the selected objects are deleted. Unselected objects, if any, are not deleted by this scrub. However, if nothing is selected, all objects touched by the scrubbing gesture are deleted.

Your application doesn't have to do anything to handle scrubbing in input areas that are based on Newton prototypes. If you make other input areas, you must program them to recognize the scrubbing gesture, play the poof sound, and display the dissipating smoke animation sequence. Scale the smoke cloud to the size of the user's zigzag gesture, keeping the height and width proportional to the original image's dimensions. Do not scale the smoke cloud down so much that it becomes unrecognizable, even if the zigzag gesture is very small.

Joining Words

To join words, a user draws a V between them at their baselines, as shown in Figure 6-20.

Figure 6-20 Joining two words

your suit cases are much
your suitcases are much

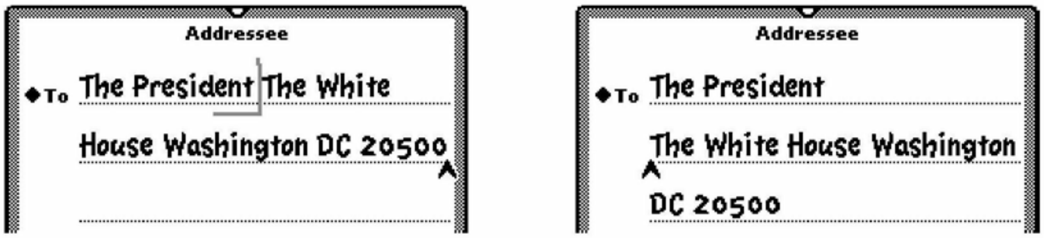
1. User draws a small V between two words

2. System joins the words

Breaking Paragraphs

To break one paragraph into two, a user draws a backwards L at the desired breaking point, as shown in Figure 6-21.

Figure 6-21 Breaking a paragraph into two paragraphs


Addressee
◆To The President The White
House Washington DC 20500
Addressee
◆To The President
The White House Washington
DC 20500

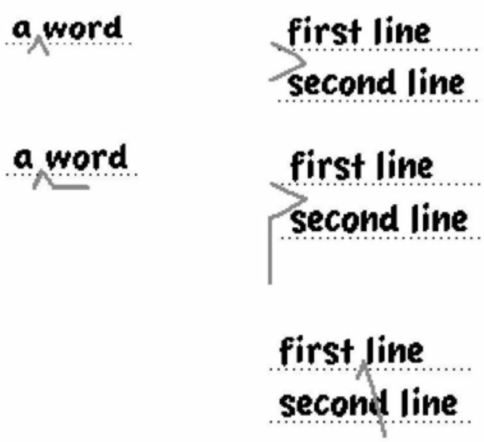
1. User draws a backwards L

2. System breaks paragraph at that point

Inserting Space in Text

To insert space in text, users draw carets and lines as shown in Figure 6-22. The top of the caret should line up with the baseline of the letters.

Figure 6-22 Inserting space in text



Inserts space for one letter

Inserts space for a single line

Insets space for a word (the longer the line, the bigger the space)

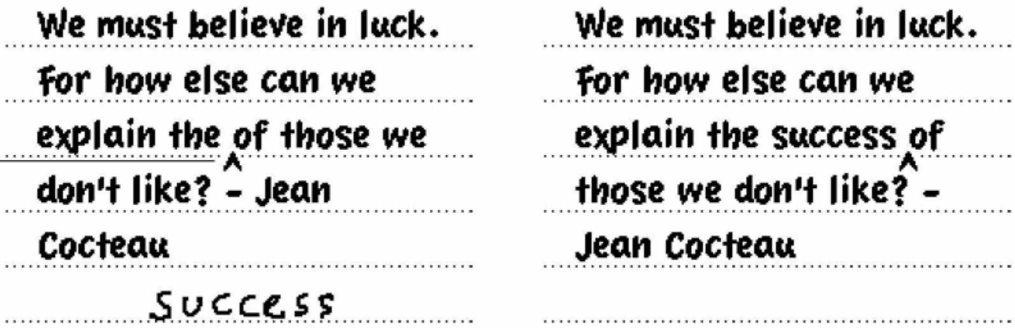
Inserts space for several lines (the longer the vertical line, the bigger the space)

Breaks the line and inserts space for several lines

Inserting New Text

When a caret is displayed in an input area, it marks the point where the Newton system will insert newly written words. No matter where a user writes in the input area, the Newton system inserts the text at the caret. It doesn't matter whether the system is set for text recognition or ink text. Figure 6-23 illustrates the caret.

Figure 6-23 A caret marks the text insertion point



Caret

1. User writes the word *success*

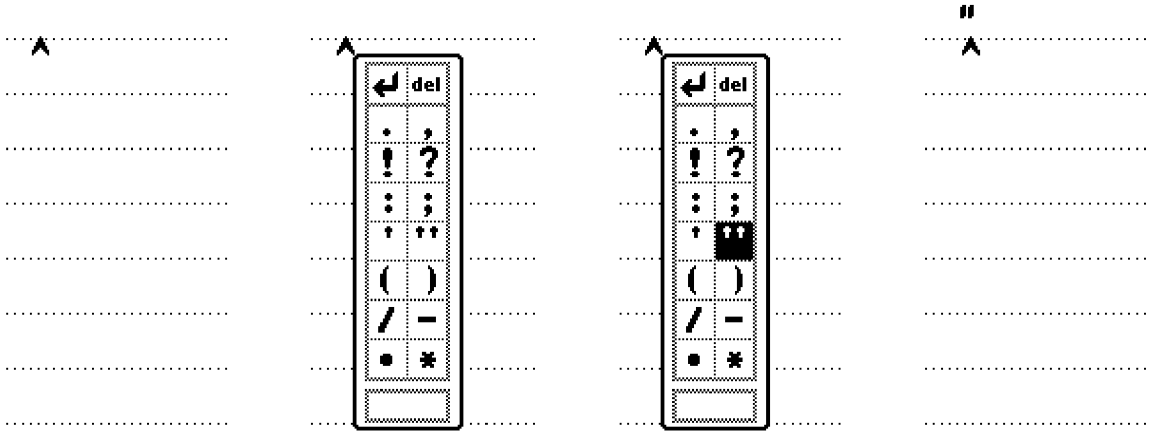
2. New word is inserted at the caret

A user can move the caret simply by tapping the screen at the desired location. Users always know and control exactly where their writing goes.

The caret is not displayed if a user turns off the "Insert new words at caret" option in the Handwriting Recognition section of the built-in Preferences application. In that case, a user can insert a new word by writing it on top of the word it should precede; the Newton system shifts the old word to the right and inserts the new word.

A user can input a punctuation mark or other special character by choosing it from the picker that pops up when the user taps the caret. In addition, the Caret picker allows a user to break a paragraph at the caret (moving the caret to the start of the new paragraph); delete the character to the left of the caret; or insert a blank space. Figure 6-24 shows the Caret picker.

Figure 6-24 The Caret picker lists 14 hard-to-write characters and three actions



1. User taps caret

2. Picker pops up

3. User selects a character

4. Selected character is inserted

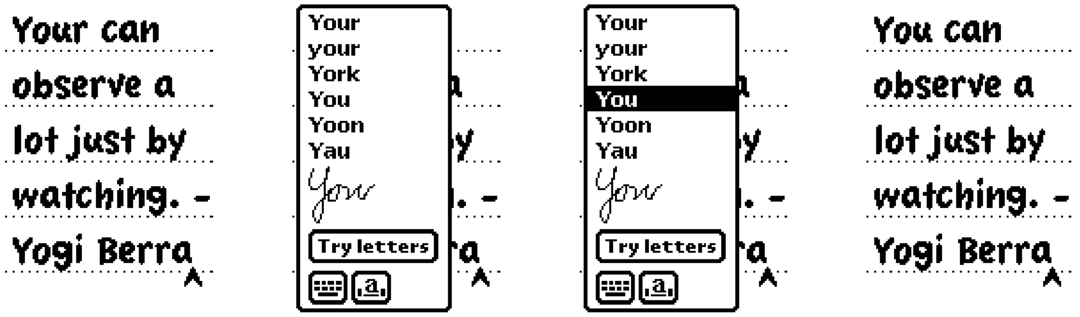
Replacing Text

By extending the method for inserting text, a user can replace existing text. Instead of tapping to position the caret, the user drags the highlighting mark to select the text to replace. Then the user writes the replacement text anywhere on the screen and the selected text is replaced.

Correcting Misrecognized Text

If the Newton system does not recognize a word correctly, the user can correct it by several means. For one, the user can replace any letter by writing another letter over it. The user also has the option of selecting or erasing the word and writing it again. Another alternative: the user can double-tap a word to pop up a picker that lists some alternate words. From the list of alternates the user can select one as a replacement. Figure 6-25 shows how a Correction picker works.

Figure 6-25 How a Correction picker works



Your can observe a lot just by watching. - Yogi Berra

You can observe a lot just by watching. - Yogi Berra

1. User double-taps the misrecognized word *Your*

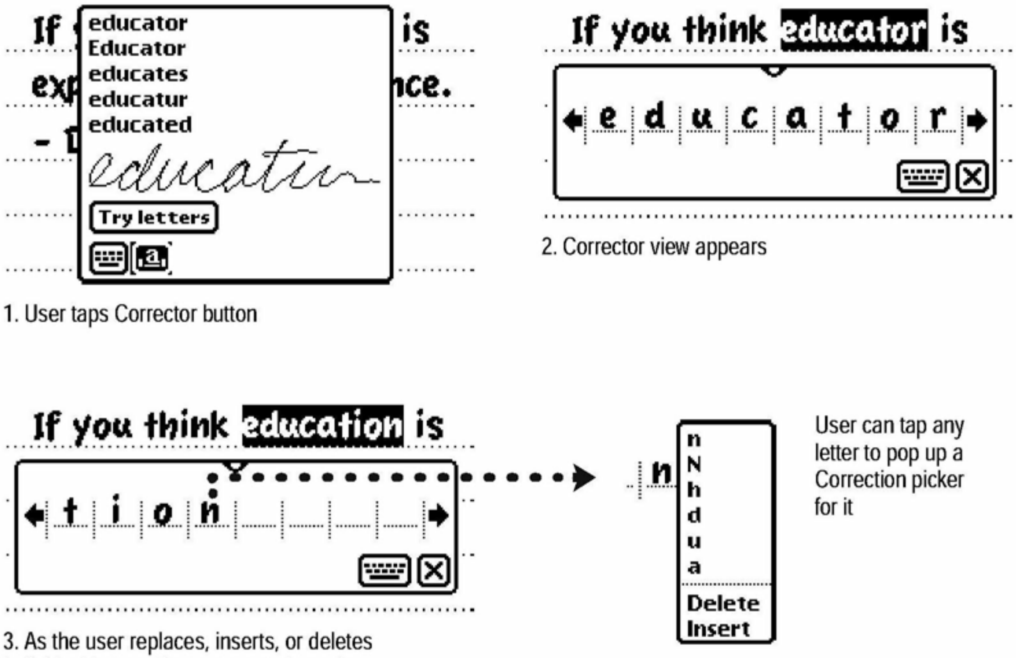
2. Correction picker pops up

3. User selects alternate word

4. Selected alternate word replaces misrecognized word

At the bottom of the Correction picker are three buttons: a Keyboard button, a Corrector button, and a Try Letters button. Tapping Try Letters causes the Newton system to ignore the recognition dictionaries and try to recognize the word letter by letter. Tapping the Keyboard button displays a keyboard, with which the user can type corrections. Tapping the Corrector button brings up a Corrector view, in which the user can make corrections to individual letters. The user can write over a letter to replace it, delete a letter by scrubbing it, or insert a space in which to write an additional letter. In addition, the user can tap a letter in the Corrector view to pop up a Correction picker that lists alternate letters plus the commands Insert and Delete. Figure 6-26 shows how a Corrector view works.

Figure 6-26 How a Corrector view works



1. User taps Corrector button

2. Corrector view appears

3. As the user replaces, inserts, or deletes individual letters, the system updates the recognized text

User can tap any letter to pop up a Correction picker for it

The Corrector view can also be used for bulk correction. In this scenario, the user leaves the Corrector open and one by one taps words to be corrected. Software developers can define corrector templates for different types of data (dates, times, etc.).

Changing Capitalization of Text

To change how a word is capitalized, a user selects the word and then draws a vertical line over it. Drawing the line in an upward direction over the first letter of the word capitalizes that letter. Drawing the line upward over the middle of the word capitalizes all letters. Drawing the line downward changes capital letters to lowercase.

Changing Paragraph Margins

To change paragraph margins, a user selects the paragraph by drawing a selection box around it. Then the user holds the pen on the left or right side of the box and drags it.

Removing Extra Space from Paragraphs

To remove extra space from a paragraph, a user selects the paragraph by drawing a selection box around it. Then the user taps the border of the box.

Duplicating Text or Shapes

After selecting text or shapes, a user can duplicate the selection by quickly tapping twice inside the selection and not lifting the pen after the second tap. That gesture is called **tap-and-a-half** because the pen goes down, up, and down again (but not up again). Keeping the pen down, the user then drags a duplicate away from the original item.

Changing Shapes

After drawing a selection box around a whole shape, a user can drag any corner to change the shape. Dragging a corner or edge of the selection box stretches, shrinks, or distorts the whole shape.

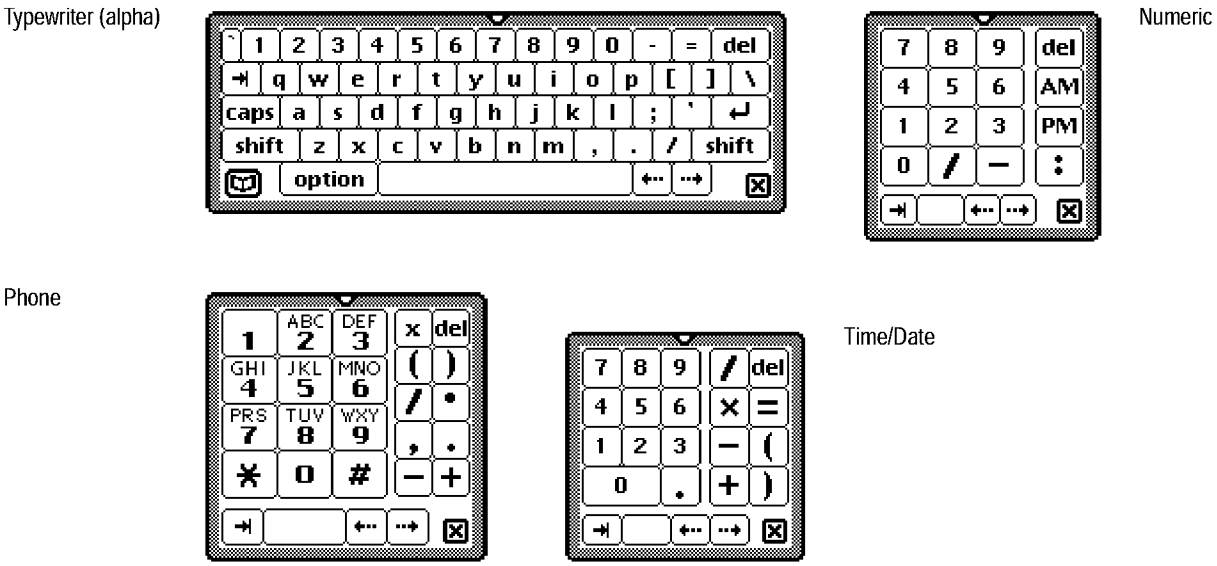
Moving Objects

A user can move an object—text, ink text, sketch, shape, or a combination of them—by selecting the object and then dragging it to another part of the same input area or to another visible input area. The user can also drag the selected object to the top, left, or right edge of the screen, where it becomes a miniature and attaches itself to the edge of the screen. Then the user can go to another view and drag the miniature from the edge of the screen to any visible input area.

Typing

Wherever users can write text on a Newton device, they can type it using an on-screen keyboard as well. Figure 6-27 shows the four keyboards that are built into the Newton system. You can also create custom keyboards for use in your application.

Figure 6-27 The four built-in keyboards


Typewriter (alpha)

Numeric

Phone

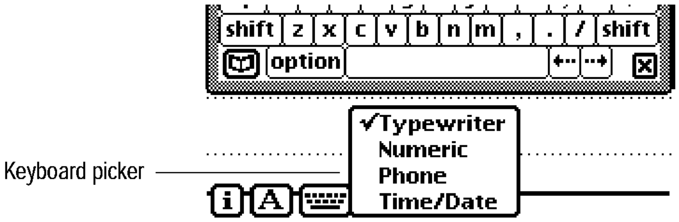
Time/Date

Displaying Keyboards

There are several ways users can display a keyboard. One is to double-tap any blank space in a text-input area. Another is to double-tap a word to bring up a Correction picker and then tap the keyboard in that picker (as described under "Correcting Misrecognized Text" on page 6-29). Users can also bring up a keyboard by tapping any visible Keyboard button. (An application can put a Keyboard button on the left side of its status bar and at the bottom left corner of slips, as described in "Keyboard Button" on page 3-25.)

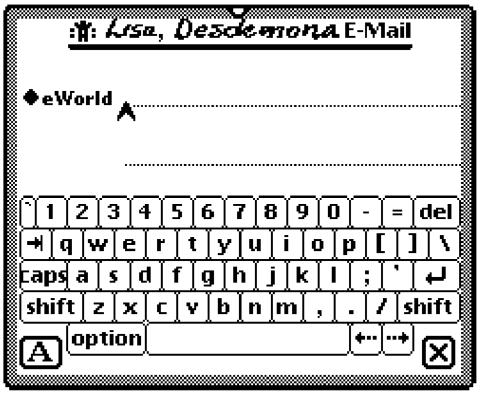
If a keyboard is already displayed, then tapping a Keyboard button pops up a Keyboard picker. The Keyboard picker lists the available keyboards, and a user can select one by tapping its name in the Keyboard picker. A check in the Keyboard picker marks the currently selected keyboard. Figure 6-28 shows the Keyboard picker in the built-in Notepad application.

Figure 6-28 A Keyboard picker lists alternate on-screen keyboards



Any time the user is about to input data that the Newton system is unlikely to recognize, such as an e-mail address, your application should automatically display an appropriate keyboard for the user. Another option is to embed a keyboard in a slip used for data input, as the built-in Names File application does. Figure 6-29 shows a keyboard embedded in a slip.

Figure 6-29 A keyboard can be embedded in a data-input slip


Lisa, Desdemona E-Mail
eWorld
` 1 2 3 4 5 6 7 8 9 0 - = del
q w e r t y u i o p [ ] \
caps a s d f g h j k l ; '
shift z x c v b n m , . / shift
option

Keyboard Position

When a user brings up a keyboard it should appear centered above the status bar, floating above other views. If possible, the keyboard should be situated vertically where it does not cover the text-insertion caret. A user can move a keyboard by dragging its drag handle, and can bring up other views above the keyboard.

Keys

There are three kinds of keys on a Newton keyboard: character keys, modifier keys, and editing keys. A character key enters text. A modifier key redefines the most of the character keys. An editing key moves the text-insertion caret.

Character Keys

Most of the keys on a keyboard are character keys. There are character keys for letters, numbers, punctuation, and blank space. If the user taps one of these keys while entering text, the corresponding character is added to the text. The tab, return, and del keys also act as character keys. Although they don't add characters to the text, they do have a visible effect on the text, as described in the following paragraphs.

Return

In a field that allows entering multiple lines of text, the return key inserts a carriage return at the text-insertion caret. It ends the current paragraph and moves the caret to the beginning of the next line. In a field that allows entering one line of text, the return key moves the caret to the next text-input field in the same container view. Tapping the return key is never a shortcut for tapping a button or other control. In particular, tapping the return key does not close the frontmost container view.

Tab

In a field that allows entering multiple lines of text, the tab key moves the text caret to the next tab stop. In a field that allows entering one line of text, the tab key moves the text caret to the next text-input field in the same container view.

Del

The del key deletes currently selected text. If no text is selected, tapping the del key removes the character preceding the text caret.

Shift

The shift key acts like its namesake on a typewriter. It changes the character produced by the next tap on a character key, making alphabetic keys produce capital letters, number keys produce punctuation or symbols, and so on. Unlike the shift key on typewriter or personal computer, the Newton shift key is not pressed concurrently with pressing a character key. The Newton shift key locks on when tapped and releases automatically when a character key is tapped.

Caps

The caps key makes the alphabetic keys produce capital letters but has no effect on any other keys. In other words, even when caps is on, a user must tap the shift key to produce punctuation and symbols with the number keys and other nonalphabetic keys. The caps key locks on when tapped and stays on until tapped again; even closing a keyboard does not turn off the caps key.

Option

The option key changes the character produced by the next tap on a character key to produce a set of international characters and special symbols. For example, in many Newton fonts, option-4 produces the ¢ symbol, option-r produces ®, and option-g produces ©. Like the Newton shift key, the Newton option key is not pressed concurrently with pressing a character key. The Newton option key locks on when tapped and releases automatically when a character key is tapped. Option can be used together with shift, in combination with a character key, to produce still other symbols. For example, option-shift-? produces the Spanish ¿ character.

Arrow Keys

The left arrow and right arrow keys move the text caret left or right one character at a time. If a user selects some text and then taps an arrow key, the text is deselected and the caret appears at the right or left edge of the selection. This action doesn't move the selected text.

If a user selects a shape and then taps an arrow key, nothing happens. The selected shape does not move, and the selection does not change.

Arrow keys never duplicate the function of scroll arrows.

The modifier keys—caps, shift, and option—have no effect on the arrow keys.

Type-Ahead and Auto-Repeat

If a user types more quickly than the Newton system can handle, the system queues the extra keystrokes for later processing. This queuing is called **type-ahead.** There's a limit to the number of keystrokes that can be queued, but this limit is usually not reached unless the user types while an application is performing a lengthy operation.

When a user holds the pen on a key for a certain amount of time, the system acts as if the user were repeatedly tapping that key. This feature, called **auto-repeat,** affects character keys and modifier keys alike. Auto-repeat does not function during type-ahead. It operates only when the Newton system is ready to accept typing.

Error Handling

Applications need to strictly check user input for errors while providing several easy ways for users to correct their mistakes.

Error Correction

Users can edit their input with a common set of gestures (see "Editing" on page 6-21). In addition, tapping the Undo button reverses the effect of a user's most recent action. Tapping Undo a second time undoes the undo.

The Newton interface elements provide undo capabilities for most user input. This includes writing, drawing, typing, correcting, editing, selecting a radio button, tapping a checkbox, setting a slider, and choosing from a picker of a labeled input line or expando.

Your application provides all other undo capabilities. For example, if a user chooses an item from an ordinary picker (not one that's part of a labeled input line or expando), your application is responsible for letting the user undo the choice by tapping the Undo button. Undo should apply to a single recent action, not to a set of actions. Users should be able to undo individual actions taken in a slip, but once the slip is closed there is nothing to undo.

You don't need to enable undo for every user action. In general, you should enable undo for actions that change data. You generally do not need to enable undo for actions that change the view of data or the environment, such as scrolling. From a user's standpoint, the most desirable actions to have reversed by the Undo button are the actions that would be most difficult to reverse manually. You should consider the needs of your audience when deciding which actions can be undone.

When a user initiates an action that can't be undone and could be very difficult to reverse by hand, your application should warn the user and give the user a chance to cancel the action. For example, if a user is about to change a lot of text with a search-and-replace operation that can't be undone, display a confirmation slip that says something to the effect of, "OK to make this change? (Can't undo)."

Error Detection

Users may notice and correct some input errors, but your application should also check input items for validity. There are a couple of approaches you can take to error checking, depending on the circumstances and users' expectations.

One approach is to check an input item for errors as soon as a user moves on to another input item. With this approach, a user must correct an error before the input gets stale or leads to other input errors. However, a user who makes lots of mistakes may feel pestered by what seem to be nitpicking error messages.

Another approach is to check all input items in one view at the same time, when a user taps a button to confirm all the input items and close the view. This approach only disrupts the user once per view instead of once per input item. If you take this approach, try to make your error messages as specific and diagnostic as possible. It's all too easy to make error messages vague.

CHAPTER 7

Routing and Communications

The Newton system provides a standard user interface for sending and receiving data by several communications methods, called **transports.** Most Newton systems come with transports for printing, faxing, beaming, and e-mailing. You can develop additional transports that users can install and remove at will, independent of installing and removing applications. The system makes newly installed transports available immediately in all applications, built-in and otherwise, and makes newly removed transports unavailable immediately in all applications.

If you are developing an application and want to give users access to transports for sending your application's data items, you include Action buttons in the application's container views. If you want your application to receive incoming data items, you have your application tell the system which types of items it can handle. The rest of the user interface for sending and receiving data items is provided by the transports, the system, and the built-in In/Out Box application.

This chapter describes the Newton routing and communications interface in detail, covering the following topics:

- What role the In/Out Box application plays in routing incoming and outgoing data items
- How users route outgoing items, including how the Action button works and how transports get routing information from users

- How users route incoming data items
- When and how transports should display status information
- When and how transports should allow users to stop an ongoing transfer of data items
- How transports should provide user preference settings
- What alternative routing methods are available

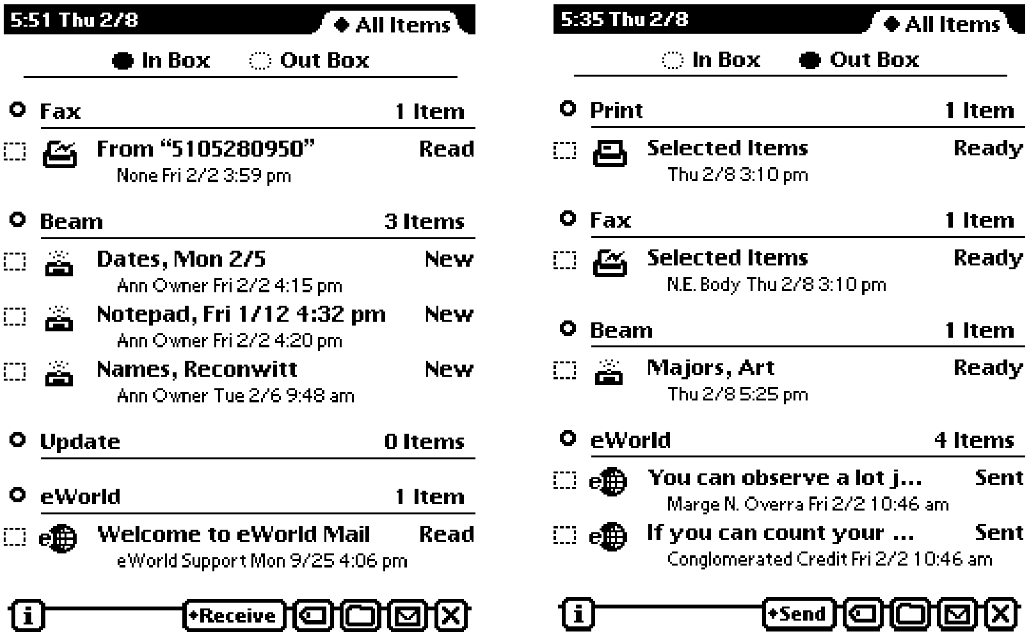
This chapter discusses the Newton routing and communications interface in the context of the applications and the transports that come with most Apple MessagePad models. Applications and transports you develop should follow these models.

The In/Out Box

The built-in In/Out Box application holds incoming and outgoing data items received or sent by a Newton transport—a communications method such as printing, faxing, beaming, and e-mailing. On some Newton devices, the In/Out Box application has two icons in the Extras Drawer, one labeled In Box and the other labeled Out Box.

A user can open the In/Out Box application by tapping whichever of these icons is present: the In Box icon, the Out Box icon, or the In/Out Box icon. Once the application is open, the user sees either incoming items in the In Box or outgoing items in the Out Box, and can switch between them by tapping a radio button in the application's main view. The user can choose to sort the items in the In Box and the Out Box by various criteria, such as date, type of transport, or status. Figure 7-1 shows the In Box and Out Box overviews with the items sorted by type of transport.

Figure 7-1 The In/Out Box application displays either the In Box or the Out Box



5:51 Thu 2/8

○ Fax 1 Item
- From "5105280950" Read — None Fri 2/2 3:59 pm

○ Beam 3 Items
- Dates, Mon 2/5 New — Ann Owner Fri 2/2 4:15 pm
- Notepad, Fri 1/12 4:32 pm New — Ann Owner Fri 2/2 4:20 pm
- Names, Reconwitt New — Ann Owner Tue 2/6 9:48 am

○ Update 0 Items

○ eWorld 1 Item
- Welcome to eWorld Mail Read — eWorld Support Mon 9/25 4:06 pm

5:35 Thu 2/8

○ Print 1 Item
- Selected Items Ready — Thu 2/8 3:10 pm

○ Fax 1 Item
- Selected Items Ready — N.E. Body Thu 2/8 3:10 pm

○ Beam 1 Item
- Majors, Art Ready — Thu 2/8 5:25 pm

○ eWorld 4 Items
- You can observe a lot j... Sent — Marge N. Overra Fri 2/2 10:46 am
- If you can count your ... Sent — Conglomerated Credit Fri 2/2 10:46 am

The In Box

The In Box is where a user first sees and deals with incoming faxes, e-mail, beamed items, and other data items received by Newton transport software. Users can view many types of incoming data items in detail while the items are still in the In Box, and can send some items directly from the In Box. Most items remain in the In Box until a user puts them into other applications. For example, a user can read incoming e-mail messages and then print them, send replies to them, or fax copies of them directly from the In Box before putting them into the Notepad application.

Some incoming items may not remain in the In Box for a user to put away. An application can have the Newton system immediately transfer specific types of incoming items from the In Box to the application. For example, incoming stock quotes from a wireless modem could be transferred automatically to a stock-tracking application.

The Out Box

The Out Box holds data items coming from all applications and waiting to be printed, faxed, beamed, e-mailed, or sent by other Newton transport software. Items in the Out Box stay there until a user physically connects the Newton to a suitable output device and chooses to send the items. For example, a user may choose to fax and e-mail several items while aboard an airplane. Those items go into the Out Box and wait until the user connects the Newton to a fax modem and a phone line and sends the items.

Users can view many types of outgoing items while the items are still in the Out Box. In addition, users can change the content, routing, or addressing of some items in the Out Box. For example, a user could direct a printed item to a different printer, edit the text of an e-mail message, or change a fax number.

Users can also send some Out Box items directly from the Out Box. For example, if the Out Box contains a fax waiting to be sent, a user can also print the waiting fax directly from the Out Box.

A transport can be designed to connect and transfer out data items a user sends to it as soon as those data items appear in the Out Box. For example, a transport for wireless communications could automatically transfer out items as soon as a user sent them from an application, without the user having to open the Out Box.

In/Out Box Items

The In/Out Box shows the same header information for each item it displays in the In Box or Out Box. Each item's header consists of an icon that identifies the transport, followed by the item's title and status. A second line shows the name of the sender or recipient, followed by the date and time the item was put in the In/Out Box. Table 7-1 explains the meanings of the standard status words displayed in the In/Out Box headers.

Table 7-1 Meanings of status words in the In/Out Box headers

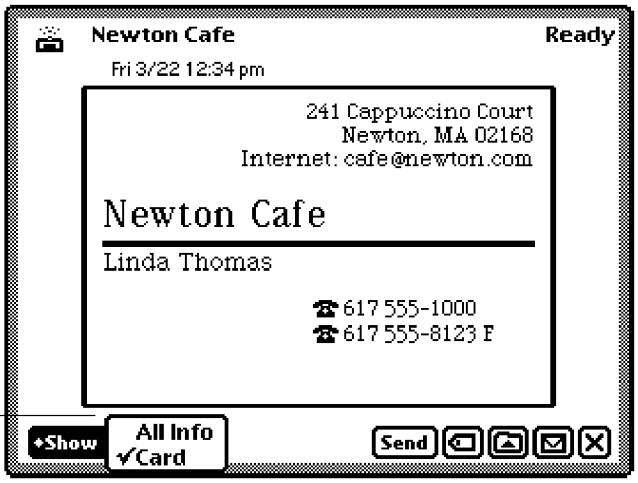
Status	**Meaning**
In Box	
New	Body has not been displayed yet
Read	Body has been displayed
Remote	Body has not been received yet (just header)
Logged	Item logged (header kept; body deleted)
Out Box	
Pending	Item not ready for sending (routing slip incomplete)
Ready	Item ready for sending
Sent	Item sent
Logged	Item logged (header kept; body deleted)
Error	Attempt to send failed

Viewing Items in the In/Out Box

Users can see more than just header information for some types of items in the In/Out Box. For example, the In/Out Box can show a page preview of print and fax items. It can also show item detail in views based on templates that other applications provide. The built-in applications provide view templates for their data, and your application can provide additional view templates for that data or its own data. If you want users to be able to see your application's data in detail in the In/Out Box, your application must define view templates and register them with the system. If no application provides a view template for a particular type of data, the In/Out Box displays a generic blank view.

If applications provide multiple view templates for the type of data currently on display in the In/Out Box application, the In/Out Box includes a Show button and picker, so users can choose among the available views. For example, when a Names File item is displayed in the In/Out Box, the Show picker lists at least the two choices All Info and Card (which the Names File application provides). Figure 7-2 illustrates a Show button in the In/Out Box.

Figure 7-2 A Show button provides access to alternative views



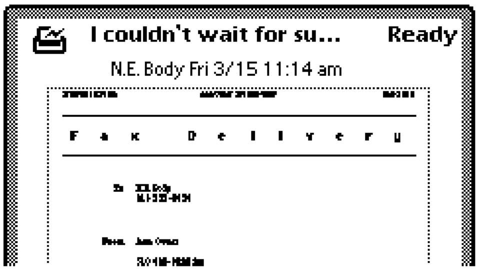
Show picker lists available views for the displayed data item

Viewing Routing Information

When viewing the detail of an item in the In/Out Box, the transport icon to the left of the item title is an Item Info picture button, and a user can tap it to display a view that gives routing information about the item. Figure 7-3 shows an example.

In the In/Out Box application, an Item Info slip displays the item title as an editable field, the transport icon and name, and the item's size. The slip may display other information, depending on the type of transport. For example, Item Info slip for a fax transport shows the fax resolution and the recipient's fax number.

Figure 7-3 Viewing routing information in an Item Info slip

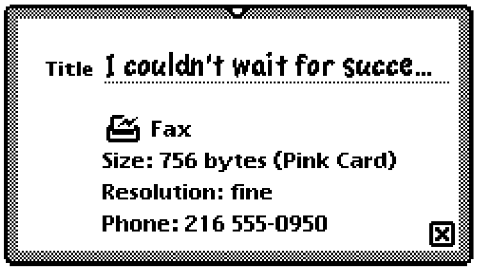


I couldn't wait for su... Ready

N.E. Body Fri 3/15 11:14 am

Fax Delivery

1. User taps Item Info button (transport icon)



Title I couldn't wait for succe...

Fax

Size: 756 bytes (Pink Card)

Resolution: fine

Phone: 216 555-0950

2. Item Info slip appears

Routing Outgoing Items

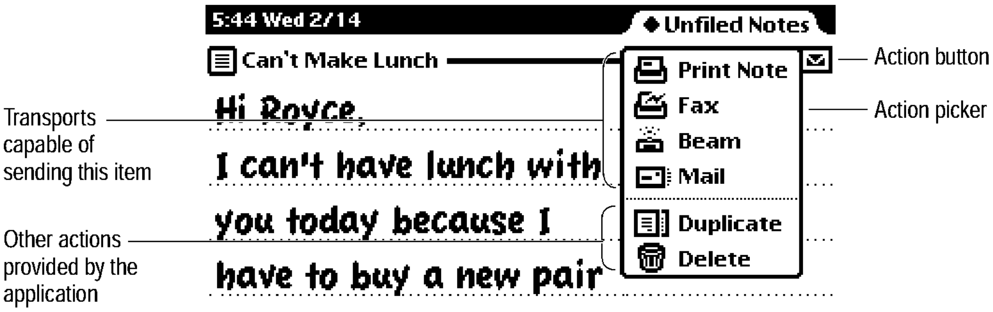
There are several steps involved in sending an item from an application through the Out Box to an output device. First, a user chooses a routing action from an Action picker. Next, the user supplies routing information in a routing slip provided by the transport that performs the chosen routing action. Then the system places the item in the Out Box. Eventually, a transport transfers waiting items in the Out Box to the proper output devices. For example, a user might choose Print from an Action picker in the Notepad application, and in the Print routing slip might choose a printer that's not available. Later, the user would connect the printer, and the print transport would send the waiting print item to the printer.

This section describes how a user sends data items from an application with an Action picker. Alternative routing methods are covered in "Routing Alternatives" on page 7-34.

Action Button and Picker

Users can send items from any application that has an Action button, which is a picture button that looks like the back of an envelope. To send the currently selected data item, a user picks a routing action from the Action picker that pops up when the user taps an Action button. The Action picker lists all transports capable of sending the currently selected data item, and it may list other actions provided by the application. Figure 7-4 shows a sample Action button and picker.

Figure 7-4 An Action picker lists the transports available for sending



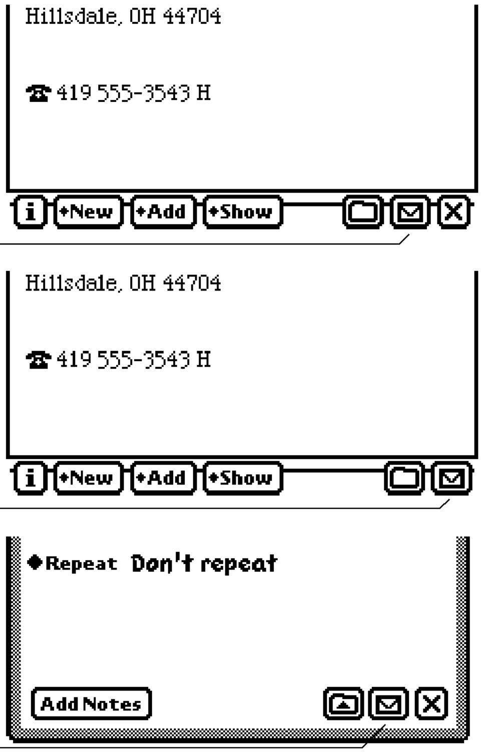
The routing action that a user chooses from an Action picker applies to the data in the view that contains the Action button. If the view contains multiple data items that can be individually selected, such as the items listed in an overview, then the routing action picked by the user applies to the currently selected items.

If there is nothing to route when a user taps an Action button (for example, if the user taps the Action button in an overview without first selecting any items), the system displays a notification alert containing the message "Nothing is selected." The warning message does not appear in an application that defines its own Action-picker items unless the application removes those items when there is nothing to act on. Your application can include actions that can function without a target item; in this case your application should not disable those actions, and the warning message will not appear.

An Action Button's Location

The scope of an Action button determines where it should be located. If an Action button can affect all the data in a view, it should go at the bottom right corner of the view, next to the view's Close box. For example, the main view of the Names File application has one Action button, and it affects all data in the view. In the backdrop application, which has no Close box, the Action button goes at the bottom right corner of the application's main view. Figure 7-5 shows examples of views with one Action button each.

Figure 7-5 An Action button at the bottom of a view affects the entire view



Hillsdale, OH 44704

☎ 419 555-3543 H

Action button on a status bar

Hillsdale, OH 44704

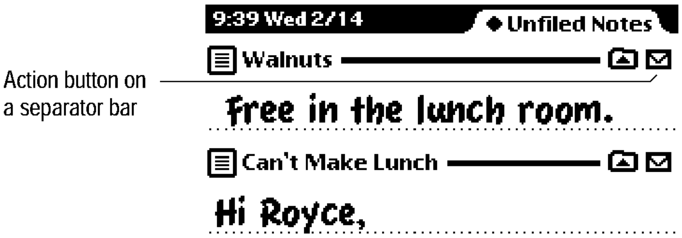
☎ 419 555-3543 H

Action button on the backdrop application's status bar

Action button in a slip

In a view where an Action button can only affect one data item of several that may be displayed (perhaps by scrolling the view), there should be an Action button above each item, at the right side of the view. Generally, such a view has a separator bar above each data item, and an Action button should be at the right end of each separator bar. For example, each note in the Notepad application has its own Action button, which applies just to that note. Figure 7-6 shows an example of Action buttons above each data item in a view.

Figure 7-6 An Action button above an item affects only that item



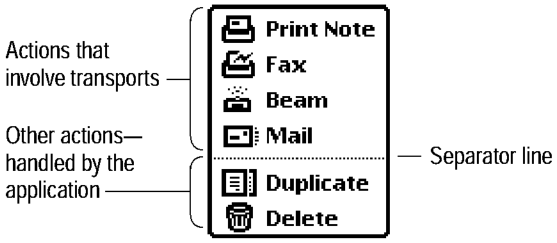
Action Picker Contents

An Action picker lists the routing actions available for the particular class of data that is currently selected. There are two types of routing actions that can appear in an Action picker:

- Routing actions corresponding to transports installed in the Newton device, such as Print, Fax, Beam, and Mail
- Application-defined actions, such as Delete and Duplicate, that do not involve the In/Out Box or a transport

Actions based on transports that work with the type of data being routed are listed at the top of an Action picker. Application-defined action commands appear at the bottom of an Action picker, below a separator line. Figure 7-7 illustrates the two parts of an Action picker.

Figure 7-7 An Action picker can include two kinds of actions


Actions that involve transports
Other actions—handled by the application
Print Note
Fax
Beam
Mail
Duplicate
Delete
Separator line

Note that the first action listed in an Action picker has the name of the target item appended to it (for example, “Print Note”). Other actions listed in the same picker do not have the name of the target item appended.

A Newton device has certain transports built in; the exact configuration depends on the capabilities of the device. For example, an Apple MessagePad 120 comes with transports for printing, faxing, e-mailing, and beaming. Users can install additional transports at any time.

Building an Action Picker

The system builds every Action picker dynamically, at the time a user taps an Action button. This allows all applications to take advantage of new transports that might be installed in the Newton device at any time. Because the system is responsible for building an Action picker, an application need not know anything about the available transports. Likewise, transports can be removed from the system without any effects on applications. The Newton system matches the transports to the data being routed, creating the Action picker on demand.

Instead of naming an individual transport, any action listed in an Action picker may name a group of related transports. For example, there might be several different e-mail transports listed as a group under the single action “Mail.” After picking the Mail action, a user would have an opportunity to select one of the available e-mail transports, as described in “Transport Picker” on page 7-18.

In addition to putting transports and transport groups at the top of an Action picker, the system puts application-defined actions at the bottom of the picker. An application can define actions that appear in all its Action pickers. It can also define a different set of actions for the Action picker in a specific view (and the views it contains).

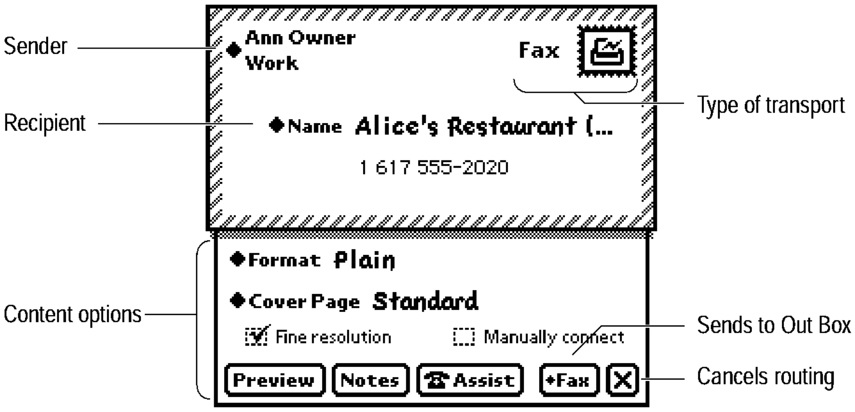
Routing Slips

When a user specifies an action that involves a transport, the transport displays a routing slip in which the user can confirm or cancel the action as well as specify additional routing information. The transport does not have to display a routing slip if a user has set transport preferences (as described in "Transport Preferences" on page 7-32) that make the routing slip superfluous. For example, the beam transport does not display a routing slip if a user has set beam preferences so that beamed items are always sent immediately and not held in the Out Box.

When a user chooses an application-defined action from below the separator line in an Action picker and the action might result in a loss of data, the application displays either a special routing slip or a confirmation alert (see "Confirmation Alerts" on page 2-18). For example, the built-in applications display a confirmation alert if a user chooses the Delete action but not the Duplicate action.

A routing slip serves the same purpose as—and even resembles—an airmail envelope and its contents. The top part of a routing slip looks like an envelope (thereby extending the Action button's visual metaphor) and contains addressing information. In the center of the routing slip envelope is the recipient, and in the upper left corner are the name and location of the sender. Where a real envelope would have a postage stamp, a routing slip displays the name of the transport and its icon in a border that looks like a postage stamp. To complete the look, the top part of a routing slip has square corners and a diagonally striped border, just like an airmail envelope. Extending below the routing slip envelope is a panel that represents what is being sent. This lower panel has controls for canceling or completing the routing action, and may have additional interface elements that affect the format and content of what is sent. Figure 7-8 shows a sample routing slip.

Figure 7-8 A routing slip shows sender, recipient, and type of transport



◆Ann Owner
Work

Fax

◆Name Alice's Restaurant (...
1 617 555-2020

◆Format Plain
◆Cover Page Standard
☑ Fine resolution □ Manually connect

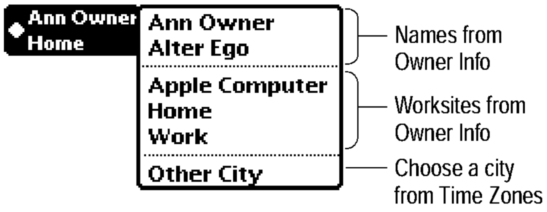
The system animates the display of a routing slip. First the envelope panel appears to slide onto the screen from the right. Then the lower panel appears to slide out of the envelope.

A routing slip is part of a transport, not part of an application. A transport uses a routing slip to get all the user-supplied information necessary to send an item. Because a transport provides the user interface for its routing slip, an application does not need to know anything about what is required to send an item via that transport.

Sender Picker

The sender's name and location displayed in the upper left corner of a routing slip are actually the label of a picker. Tapping the label pops up a Sender picker, from which a user can choose a different sender name or worksite. Figure 7-9 illustrates the Sender picker.

Figure 7-9 Changing the sender's name or location



The Sender picker lists the owner names and worksites that have been entered in the built-in Owner Info application. The last item in the Sender picker, Other City, brings up a picker from which a user can choose another city. Choosing another city for the sender in a routing slip automatically makes that city the home city in the Time Zones application.

If a user chooses a different owner, worksite, or city, the system updates the routing information as needed. For example, in a Fax routing slip the system prefixes the destination fax number with a 1 and the area code only if a user chooses a worksite or city with a different area code. The system doesn't add a 1 or the area code if the sender's worksite has the same area code as the destination.

The default owner name (or **persona,** as it is sometimes called) shown by this picker is the one corresponding to the last-used owner name for a routing operation. The default worksite for the owner is the one corresponding to the last worksite used for a routing operation (including a canceled routing operation) or the setting of the home location in the Time Zones application, whichever was done last. Note that users can create additional owner names and worksites in the Owner Info application, and they can add cities to the Time Zones application.

Some transports need the sender's return address as well as the sender's name. The transport extracts any return address information it needs, such as the sender's fax number or e-mail address, from the sender's card in the Owner Info application.

Recipient Pickers

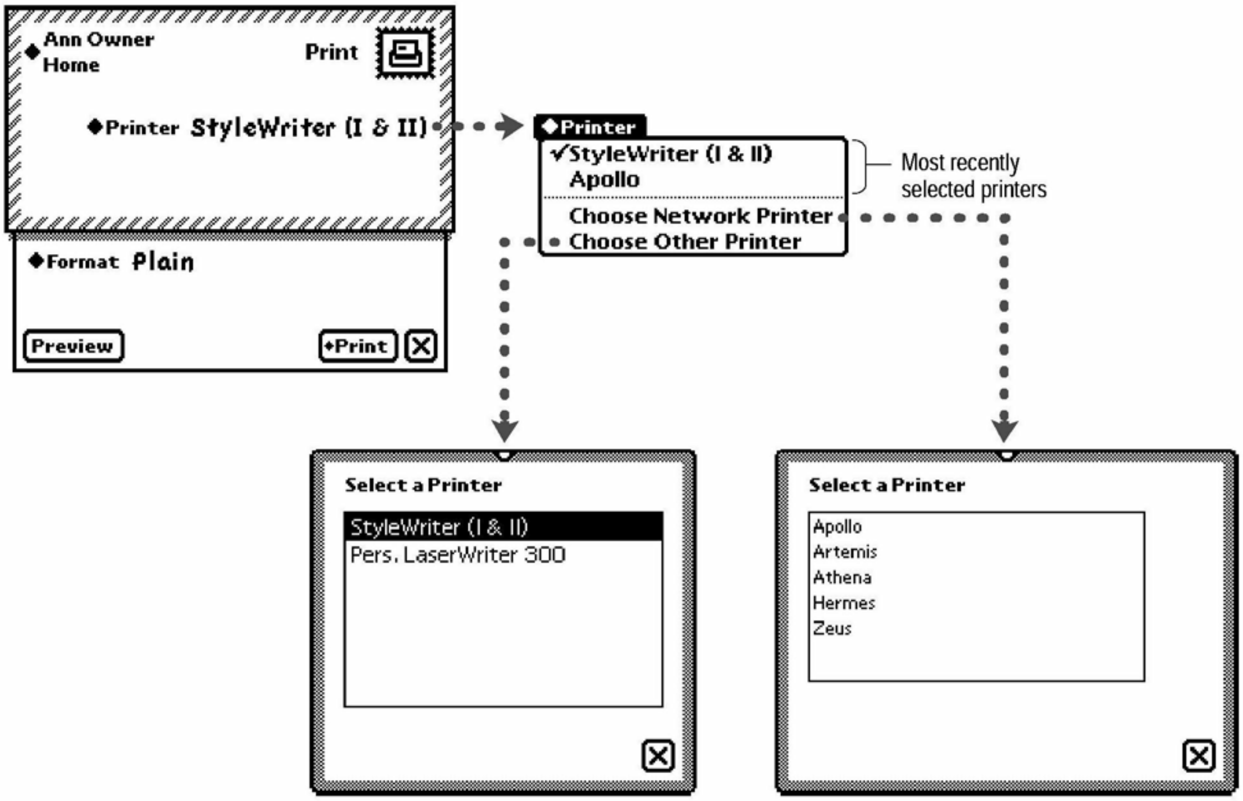
The kind of recipient information displayed in the center of a routing slip envelope depends on the kind of transport involved. For printing, the recipient is the model or name of the printer to use. For faxing or e-mail, the recipient is a name and fax number or e-mail address taken from the Names File. For beaming, the recipient is any device with a compatible infrared port.

Print, fax, and e-mail transports should display the name of the recipient as a picker label. A user can tap the name to pop up a People picker in which to select different recipients. The built-in beam transport does not need to specify a particular recipient; it displays a generic recipient that a user does not change. For a description of People pickers, see “People Picker” on page 4-27.

Choosing a Printer

When a user taps the printer named in the center of a print routing slip, a list picker pops up. The picker lists printers that a user has recently chosen, along with items that allow the user to choose a different printer. The user can choose to select a network printer or other printer. If a user chooses to select a network printer and is connected to a network, the system displays a list of printers it finds on the network. If the user chooses to select a non-network printer, the system displays a list of printers for which driver software is installed on the Newton device. Figure 7-10 illustrates the process of choosing a printer.

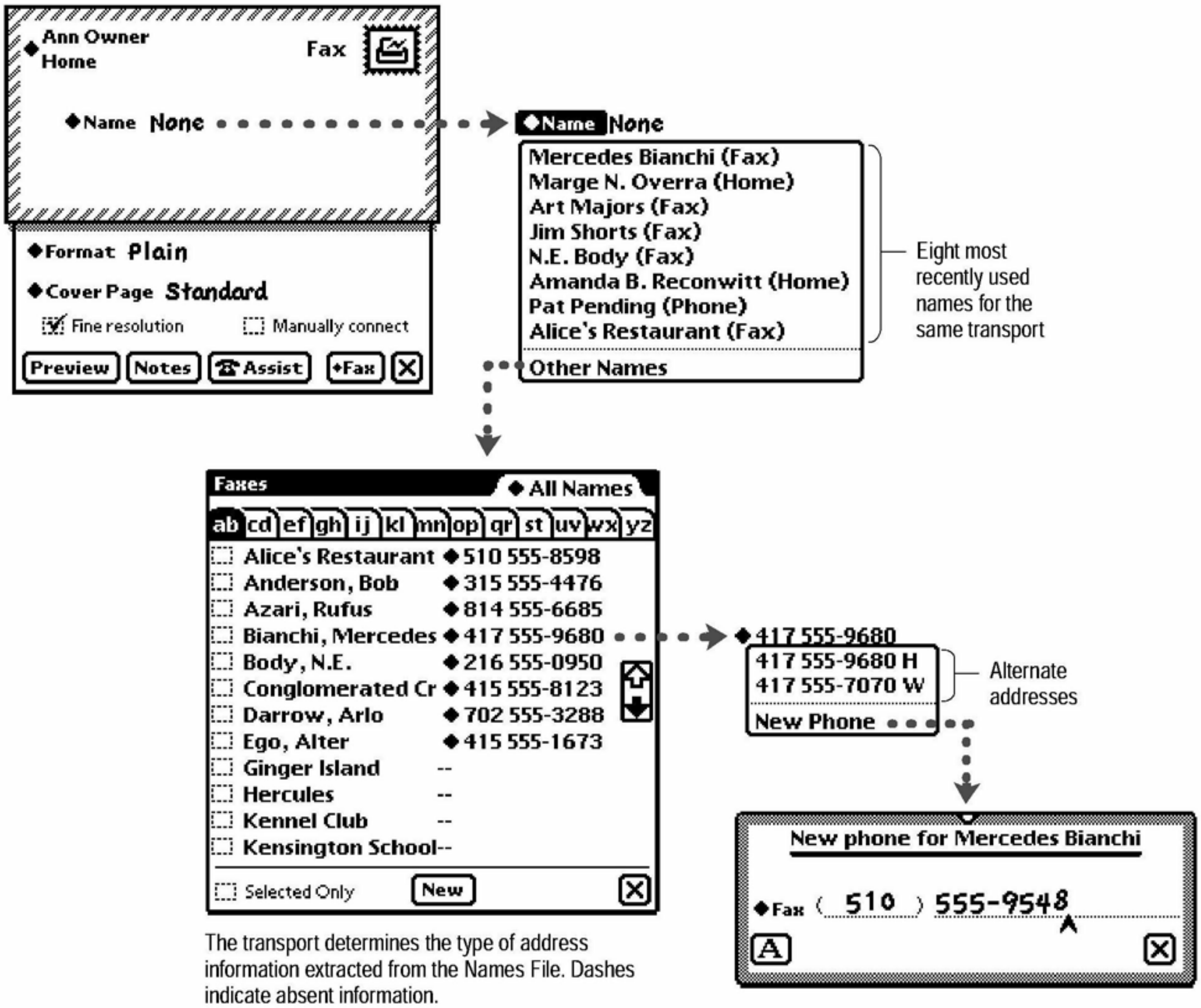
Figure 7-10 Choosing a printer in a routing slip



Choosing Fax or E-mail Recipients

The recipient in a fax or e-mail routing slip is a picker label. Tapping it pops up a picker that lists names a user has recently chosen, along with an Other Names item. The user can select a recipient from the picker list, or the user can tap Other Names to select different recipients from a People picker. A fax transport's routing slip generally has only one field for recipients, labeled Name. An e-mail transport's routing slip may have three fields for recipients, labeled To, Cc, and Bcc. The To field identifies primary recipients; the Cc (carbon copy) field identifies secondary recipients; and the Bcc (blind carbon copy) field identifies recipients whose names and addresses are hidden from To and Cc recipients. Figure 7-11 illustrates the process of choosing fax or e-mail recipients.

Figure 7-11 Choosing fax or e-mail recipients in a routing slip



◆Ann Owner
Home

Fax

◆Name None

◆Format Plain

◆Cover Page Standard

☑ Fine resolution ☐ Manually connect

◆Name None

Mercedes Bianchi (Fax)
Marge N. Overra (Home)
Art Majors (Fax)
Jim Shorts (Fax)
N.E. Body (Fax)
Amanda B. Reconwitt (Home)
Pat Pending (Phone)
Alice's Restaurant (Fax)
Other Names

Eight most recently used names for the same transport

Faxes ◆ All Names

☐	Alice's Restaurant	◆510 555-8598
☐	Anderson, Bob	◆315 555-4476
☐	Azari, Rufus	◆814 555-6685
☐	Bianchi, Mercedes	◆417 555-9680
☐	Body, N.E.	◆216 555-0950
☐	Conglomerated Cr	◆415 555-8123
☐	Darrow, Arlo	◆702 555-3288
☐	Ego, Alter	◆415 555-1673
☐	Ginger Island	--
☐	Hercules	--
☐	Kennel Club	--
☐	Kensington School	--

☐ Selected Only

◆417 555-9680

417 555-9680 H
417 555-7070 W
New Phone

Alternate addresses

New phone for Mercedes Bianchi

◆Fax (510) 555-9548

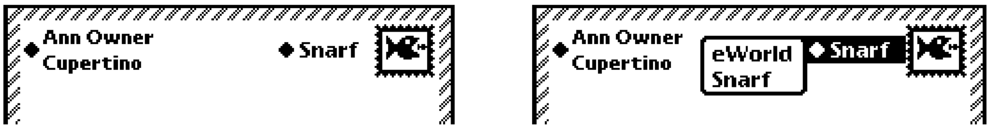
The transport determines the type of address information extracted from the Names File. Dashes indicate absent information.

The very first time a user taps the recipient in a fax or e-mail routing slip, the picker that lists recently used names does not appear because no names have been used yet. Instead, a People picker appears immediately, listing possible recipients from the Names File.

Transport Picker

A transport displays its name and icon (dressed as a postage stamp) in the upper right corner of its routing slip. If there is more than one transport for the type of routing action chosen from the Action picker, the transport name is a picker label. Tapping it pops up a picker from which a user can choose any transport in the group.

Figure 7-12 Switching to another transport in a group



1. User taps a transport name that begins with a diamond

2. Picker pops up, listing all transports of the same type

When a user switches to another transport in a group, the system closes and reopens the routing slip because each transport (not each group of transports) specifies its own routing slip. The system remembers the most recently chosen transport in a group and uses that transport if a user later chooses the same routing action from an Action picker.

Send Button and Close Box

Every routing slip has a large Close box in its lower right corner. If a user taps a routing slip's Close box, the routing slip closes with a visual effect of quickly sliding off the screen to the right. Nothing else happens.

To the left of the Close box is a text button labeled with the name of the routing action.Tapping this text button, which is known as the Send button, closes the routing slip, but with different animation than the Close box. First the lower panel slides up, as if it were going into the envelope part of the routing slip. Then the envelope slides off the screen to the right.

In addition to the routing slip closing, one of three things happens when a user taps a Send button. The outgoing item may be placed in the Out Box and sent immediately. The item may be placed in the Out Box and held there until a user sends it. Or a picker may pop up, giving the user the choice of sending now or later. Customarily a transport allows users to control which of those three alternatives happens by setting preferences in the In/Out Box application, as described in “Transport Preferences” on page 7-32.

A transport can also force the button to send now or later without displaying a picker. This overrides any preferences setting a user may have made for the transport.

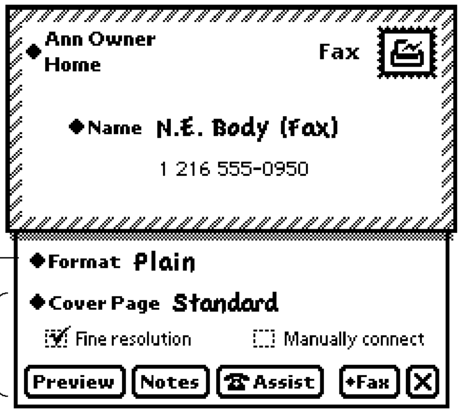
When saving items in the Out Box for later transmission, a transport has to determine when it should get information from the sender’s owner and worksite cards. Although this issue doesn’t affect the user interface visibly, the issue does affect stability and consistency as they appear to users. In general, the transport should get information from the sender’s owner card at the time the item is sent to the Out Box. Such information might include the sender’s name, return address (such as fax number or e-mail address), credit card information, and so on.

However, if the transport uses worksite information to make a connection (for example, to determine how to dial the recipient’s fax number), the transport should wait to obtain the most current information—based on the user’s current worksite setting—until the item is actually transmitted from the Out Box, and make necessary adjustments at that time. For example, if a user queued several fax items at home but didn’t send them until at work, the transport might need to change the area code information for dialing the recipient’s fax number.

Other Routing Slip Elements

A routing slip's lower panel may have additional controls and pickers that affect what is sent and how it is sent. The system includes a Format picker if there is more than one format for the class of data being sent. A transport may include additional pickers, buttons, checkboxes, radio buttons, and other interface elements that users may need to prepare the routing action. For example, the built-in fax transport includes six additional interface elements: a picker, two checkboxes, and three buttons. Figure 7-13 shows the interface elements in the lower panel of a Fax routing slip.

Figure 7-13 Setting format and content options in a routing slip


◆ Ann Owner
Home

Fax

◆ Name N.E. Body (Fax)
1 216 555-0950

◆ Format Plain
◆ Cover Page Standard
☑ Fine resolution □ Manually connect

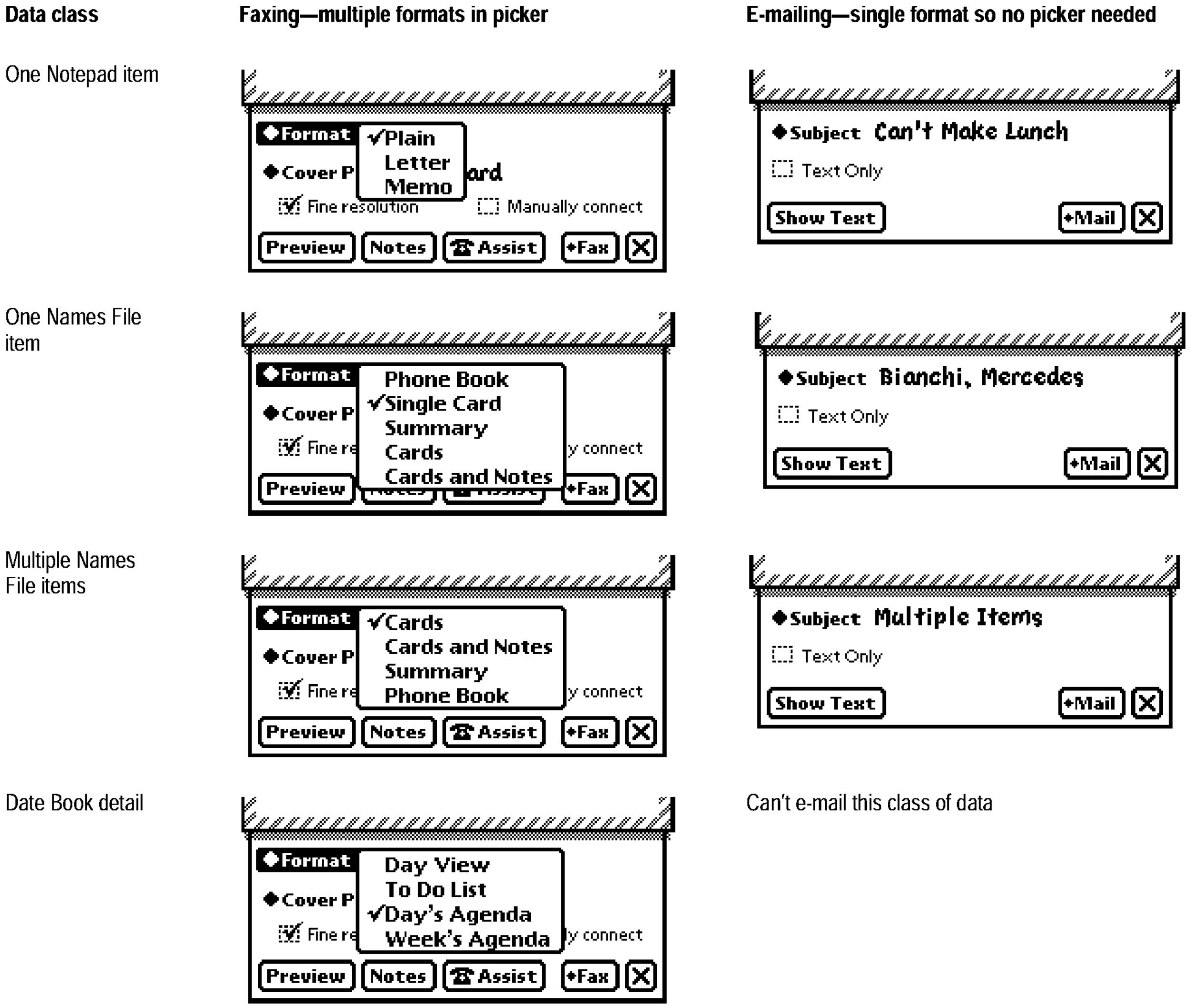
System includes Format picker if more than one format exists for outgoing data

Transport provides other user options

Format Picker

A routing slip's Format picker lists all the routing formats that apply to the outgoing item or items. For example, five formats might apply when faxing an item, while only one format applies when beaming. Figure 7-14 shows the variety of format choices available in several built-in applications on an Apple MessagePad 120.

Figure 7-14 Format choices vary by transport and class of data



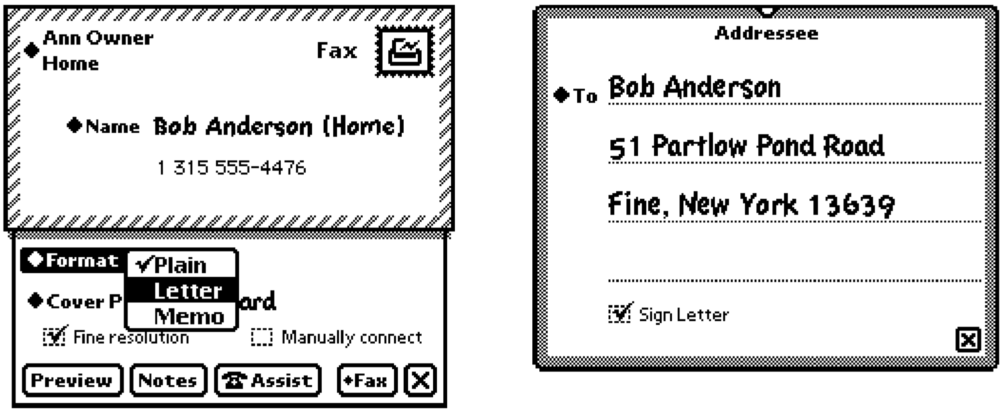
Data class
Faxing—multiple formats in picker
E-mailing—single format so no picker needed
One Notepad item
◆Format ✓Plain Letter Memo
◆Cover P ard
☑ Fine resolution □ Manually connect
Preview Notes ☎Assist ◆Fax ☒
◆Subject Can't Make Lunch
□ Text Only
Show Text ◆Mail ☒
One Names File item
◆Format Phone Book ✓Single Card Summary Cards Cards and Notes
◆Cover P
☑ Fine re y connect
Preview ◆Fax ☒
◆Subject Bianchi, Mercedes
□ Text Only
Show Text ◆Mail ☒
Multiple Names File items
◆Format ✓Cards Cards and Notes Summary Phone Book
◆Cover P
☑ Fine re y connect
Preview Notes ☎Assist ◆Fax ☒
◆Subject Multiple Items
□ Text Only
Show Text ◆Mail ☒
Date Book detail
◆Format Day View To Do List ✓Day's Agenda Week's Agenda
◆Cover P
☑ Fine re y connect
Preview Notes ☎Assist ◆Fax ☒
Can't e-mail this class of data

Although a transport specifies most items in its routing slip, it does not determine which formats to list in a Format picker. The system builds a Format picker each time it displays a routing slip, and it determines which formats to list based on the transport and the class of data being routed.

Each application defines routing formats for its classes of data and registers the formats with the system. Typically, an application defines several routing formats so that users have a choice of routing actions. For example, an application might define two formats for printing and faxing image data, one format for beaming or e-mailing structured data, and another format for e-mailing text data. If two or more applications happen to define routing formats for the same class of data, the system makes all those formats available whenever a user routes that class of data in any of the applications.

Any format can specify an auxiliary view for getting supplemental information from the user. When a user chooses that format, the system displays the auxiliary view. For example, choosing the Letter format when printing or faxing an individual note in the Notepad application brings up a slip in which the user supplies the addressee and indicates whether to include the ink text signature. Figure 7-15 illustrates that example.

Figure 7-15 A format can get supplemental information in an auxiliary view



◆ Ann Owner
Home

Fax

◆ Name Bob Anderson (Home)
1 315 555-4476

◆ Format ✓Plain
Letter
Memo

◆ Cover P…ard

☑ Fine resolution ☐ Manually connect

Addressee

◆ To Bob Anderson
51 Partlow Pond Road
Fine, New York 13639

☑ Sign Letter

1. User chooses Letter format when faxing a Notepad item

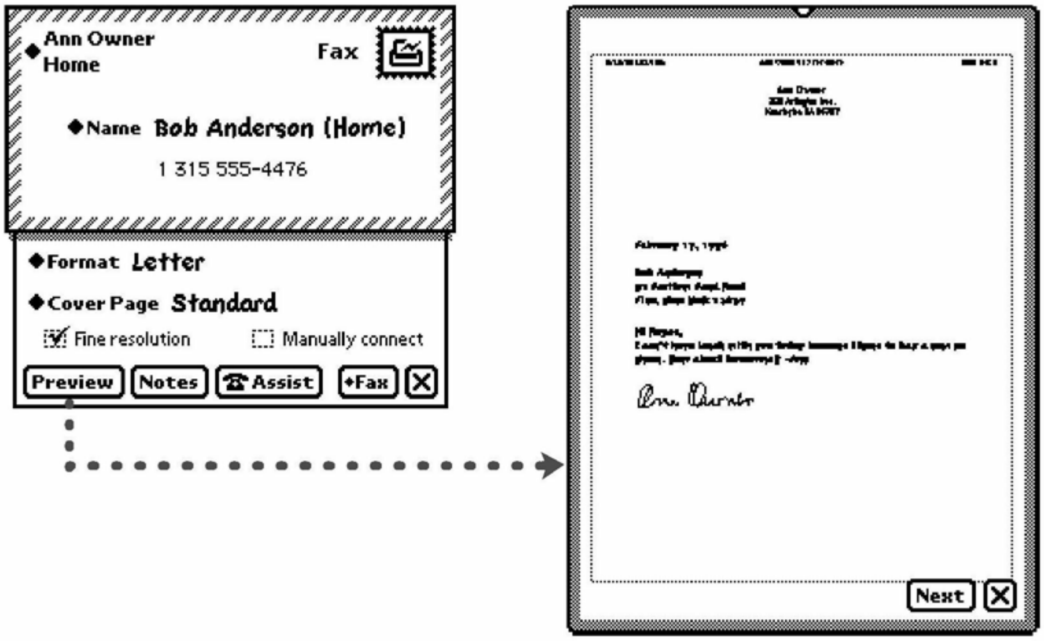
2. Letter format displays a slip requesting additional information

Each time a routing slip opens, the system initially sets the format to the format most recently used for the transport and class of data. If the class of data has never been routed through the transport before, the system makes the initially selected format the first format it finds.

Preview Button

A transport that routes page images should allow users to preview the pages from its routing slip. The built-in print and fax transports do that by including a Preview button in the lower left corner of their routing slips, aligned with the large Close box in the opposite corner. Tapping the Preview button displays a large slip containing a reduced image of one outgoing page. The slip contains a Next button for advancing to the next page and a large Close box for closing the slip. Figure 7-16 shows an example of a Preview slip.

Figure 7-16 Previewing outgoing page images



◆ Ann Owner
Home

Fax

◆ Name Bob Anderson (Home)
1 315 555-4476

◆ Format Letter

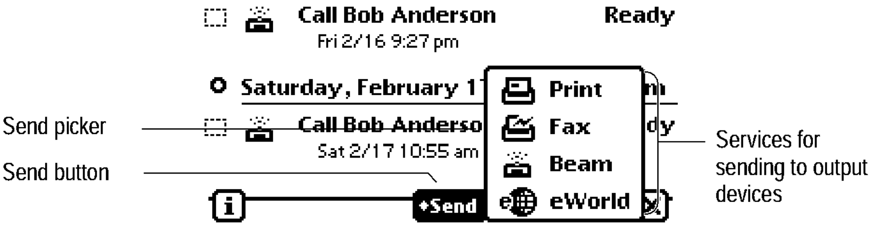
◆ Cover Page Standard

☑ Fine resolution ☐ Manually connect

Sending Out Box Items

Items a user chooses to send later (as described in "Send Button and Close Box" on page 7-18) wait in the Out Box until the user is ready to have the transports transfer the items out of the Newton device. At that time the user connects the Newton to an output device and chooses a matching output service from the Send picker that pops up when the user taps the Send button in the Out Box. For example, to send faxes waiting in the Out Box, a user connects a fax modem and chooses Fax from the Out Box's Send picker. The Send picker lists all Newton transports capable of transferring items from the Out Box to an output device. Figure 7-17 shows a sample Send button and picker in the Out Box.

Figure 7-17 The Out Box's Send picker lets users send items to output devices



Before choosing an output service from the Out Box's Send picker, a user can select specific queued items to be sent. If the user does not select items first, the chosen service sends all items waiting for it.

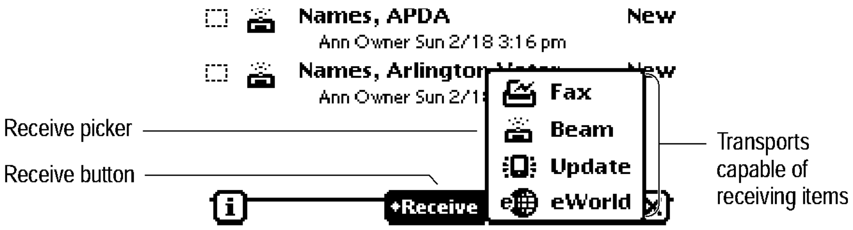
Routing Incoming Items

As mentioned earlier, users receive incoming data items through the In Box part of the built-in In/Out Box application. The items come from Newton transports that connect to sources of incoming items, retrieve items, and transfer the items to the In Box. For example, the built-in fax transport can connect to a calling fax machine, retrieve a fax, and transfer it to the Newton In Box.

Receiving In Box Items

To receive items, a user can pick a routing action from the In Box's Receive picker, which pops up when the user taps the Receive button. The Receive picker lists all Newton transports capable of receiving data items from external sources. Figure 7-18 shows a sample Receive button and picker.

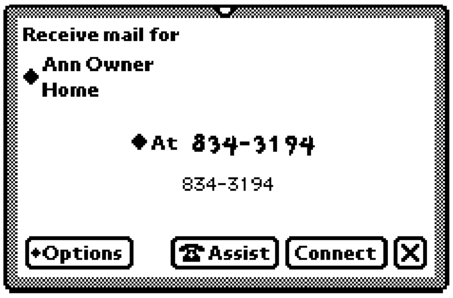
Figure 7-18 The Receive picker lists the transports available for receiving



When a user picks a routing action from the Receive picker, the corresponding transport connects to its source of incoming data. Each transport may have a different procedure for connecting. For example, the fax transport displays a slip asking whether the user wants to wait for a fax call or connect manually. An e-mail transport might display a slip in which a user sets up the phone number to dial, specifies what items to retrieve, and taps a Connect button when ready. The beam transport tries to connect immediately. Figure 7-19 shows the connection slips for the fax transport and an example e-mail transport.

Some transports can connect automatically when the system detects incoming data items for them. For example, the beam transport can connect automatically to another Newton device that is sending through its beam transport. A transport that can connect automatically should allow users to disable automatic connection by setting preferences in the In/Out Box application, as described in "Transport Preferences" on page 7-32.

Figure 7-19 Connection setup varies by transport



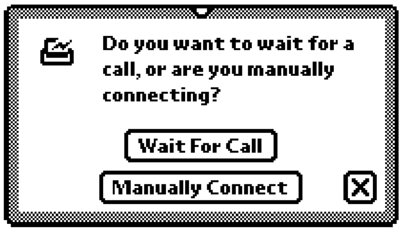
Receive mail for

◆ Ann Owner
Home

◆At 834-3194

834-3194

An e-mail transport might display this slip



Do you want to wait for a call, or are you manually connecting?

Built-in fax transport displays this slip

A transport can also allow users to schedule times when it automatically connects and receives incoming items. Users schedule connect times by setting preferences in the In/Out Box application, as described in "Transport Preferences" on page 7-32.

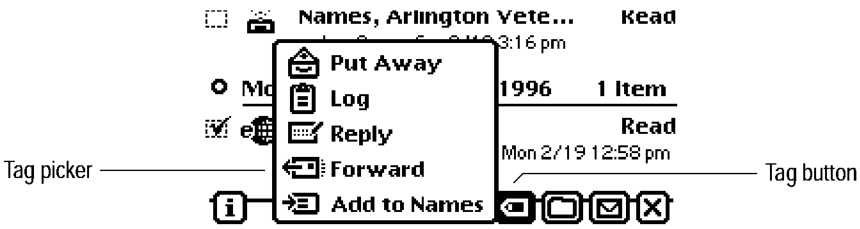
Receiving Remote In Box Items

A user may not want an e-mail transport to receive all items every time it connects, due to the length of time it would take or the amount of storage space they would occupy on the Newton device. An e-mail transport (or other transport that potentially receives numerous items from a remote source at a slow speed) can retrieve just the item headers and put them in the In Box with a status of Remote. That status tells users that the body of the item is stored remotely and has not yet been transferred to the Newton. If a user attempts to view an incomplete item in the In Box, the In Box notices the Remote status and has the appropriate transport get the remainder of the item from the remote source.

Disposing of Received Items

Most received items remain in the In Box until a user disposes of them. To dispose of In Box items, a user selects one or more of them and chooses an action from the Tag picker, which pops up when the user taps the Tag button. The Tag picker lists only actions that apply to at least one of the selected items. Figure 7-20 shows a sample Tag button and Tag picker.

Figure 7-20 The Tag picker disposes of currently selected In Box items



Putting Away Received Items

The Tag picker includes the Put Away action if an item the user has selected can be transferred from the In Box to another installed Newton application. Each application registers with the system the types of items it can accept from the In Box. For example, the built-in Notepad application accepts text items such as e-mail messages in addition to regular Notepad items.

Independent of applications' abilities to accept items from the In Box, each transport can include a method for putting away items it has received. If a transport can put away an item the user has selected, the Tag picker includes a Put Away action whether or not any applications have registered to accept that type of item.

When a user chooses to put away an item, the In Box displays a slip identifying the target application. If there are multiple targets—multiple applications that can accept the item and a transport with a put-away method—the slip includes a picker label that the user can tap to pop up a list picker from which to choose one of the targets. The slip also gives the user the option of having that item deleted from the In Box or having a copy kept there.

If no application has registered to accept an item the user has selected, and the transports that received the selected items do not know how to put away items, the Tag picker does not include the Put Away action. For example, none of the built-in applications registers to accept page-image data like faxes, and the built-in fax transport does not include a method for putting away items it receives, so the Tag picker does not include a Put Away action when a user selects only faxes in the In Box. The Tag picker does include Put Away when a user selects faxes together with some other class of data that can be put away, but the selected faxes are not put away.

Putting Away Items Automatically

The In Box can put away some incoming items automatically. That happens if an application has registered to automatically accept items designated for it. As soon as the In Box receives such an item, it transfers the item from the In Box to the application without user intervention. For example, the In Box could automatically transfer incoming stock quotes from a wireless modem to a stock-tracking application. If the In Box can't automatically put away a received item because the target application is missing (perhaps it is on a card that is not inserted), the In Box holds the item until the application is present. Then the In Box automatically transfers all items it is holding for that application. What happens to an item after the In Box automatically puts it away depends on the transport involved. The transport may have the In Box delete the item, delete the item and make a log entry in the In Box, or keep a copy of the item.

Filing Items That Are Put Away

In general, when an application gets items from the In Box it should put them away unfiled so users can find them. If an application puts away an item without regard to its folder, the item will be filed in the same folder on the receiving Newton as it occupied on the sending Newton. That could make the item hard for a user to find, especially if the folder is undefined on the receiving Newton. Your application can alleviate this problem by putting away all items unfiled even if the recipient has a folder of the same name as the sender.

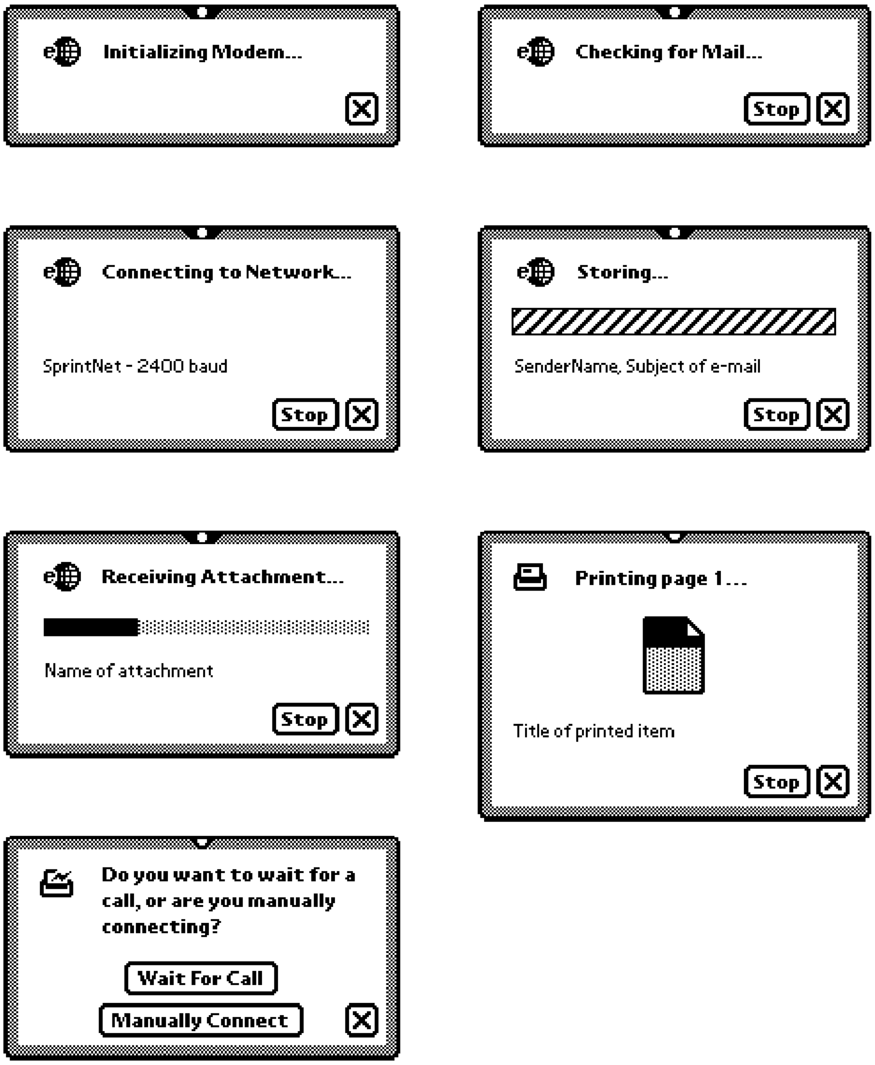
Extending the Tag Picker

A transport can add actions to the Tag picker. For example, an e-mail transport might add the actions Reply and Forward so users could reply to and forward received e-mail directly from the In Box. The built-in transports define the following action items: Reply, Forward, Copy Text to Notepad, and Add Sender to Names. There is a standard icon for each of these actions. If you create a new transport that includes these actions, use the standard icons for them.

Routing Status

A transport should display a status slip whenever it is engaged in some lengthy activity such as sending or receiving data. A transport's status slip includes a transport icon, a status message, and a large Close box. Most status slips also have a Stop button. In addition, a status slip can include the title of the item being routed and a progress indicator. If a transport needs to have a user choose between two alternatives, it can display a status slip with up to three lines of text and two buttons. The system defines several standard types of status slips, as shown in Figure 7-21.

Figure 7-21 Status slips apprise users of lengthy transport activities



Initializing Modem...

Checking for Mail...

Connecting to Network...

SprintNet - 2400 baud

Storing...

SenderName, Subject of e-mail

Receiving Attachment...

Name of attachment

Printing page 1 ...

Title of printed item

Do you want to wait for a call, or are you manually connecting?

Transports can dynamically switch from one type of status slip to another without closing the status slip, and can easily update the contents of the status slip as well (for example, updating a progress indicator).

All transports that use the standard status slips have a similar user interface and match the use of other status slips throughout the system. For general information on status slips, see "Status Slips" on page 2-20.

A status slip's Close box allows a user to hide the slip without halting the action that the slip is monitoring. If a user taps a status slip's Close box, the transport closes the slip and registers the action with the system's Notify service. The Notify service displays the flashing notify button at the top of the screen and adds the action to the Notify picker as described in "Notify Button and Picker" on page 8-2. The action continues in the background, and the user can perform another task. The user can also redisplay a status slip by choosing the corresponding action from the Notify picker. If a transport completes an action while its status slip is not displayed, the transport must unregister the action so the Notify service will remove it from the Notify picker.

A status slip's Stop button allows a user to halt the action that the slip is monitoring, as described in the next section.

Stopping a Send or Receive in Progress

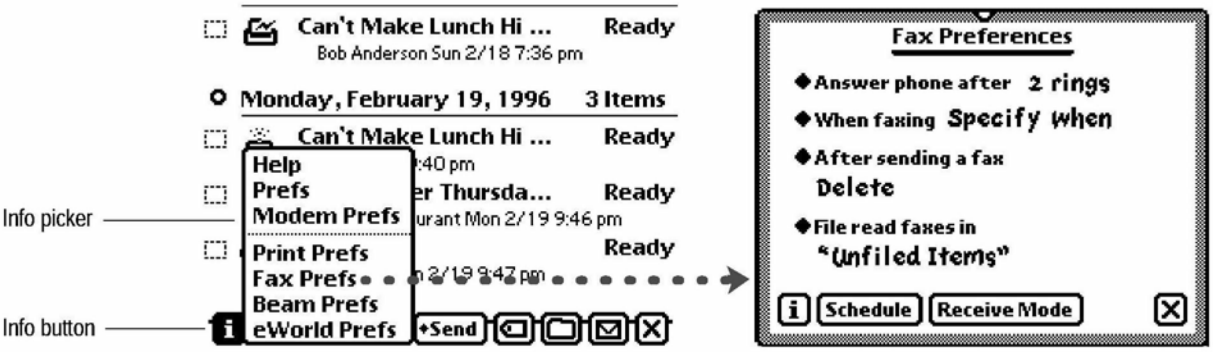
A transport should stop an ongoing send or receive operation as soon as possible under two conditions. One is when a user taps the Stop button in the transport's status slip. The second is when the system notifies the transport that it wants to turn off power.

If the system is about to turn power off while a transport is engaged (not idle), the transport should handle the situation gracefully. Generally this means displaying a confirmation alert asking for the user to confirm that it is OK to break the connection. After the user consents, the system waits for the transport to become idle before turning off the power.

Transport Preferences

The Newton system stores user-configurable preferences and other configuration information for the built-in transports, and can do the same for custom transports. The stored preferences correspond to items in a preferences slip for each transport. Users access the transports' preferences slips from the Info picker that pops up when a user taps the Info button in the In/Out Box application. Each transport that has a preferences slip is listed in the In/Out Box's Info picker. Figure 7-22 illustrates the procedure.

Figure 7-22 Accessing transport preferences from the In/Out Box's Info picker



Info picker

Info button

Can't Make Lunch Hi ... Ready
Bob Anderson Sun 2/18 7:36 pm

Monday, February 19, 1996 3 Items

Can't Make Lunch Hi ... Ready
:40 pm

er Thursda... Ready
urant Mon 2/19 9:46 pm

Ready
n 2/19 9:47 pm

Fax Preferences

◆Answer phone after 2 rings

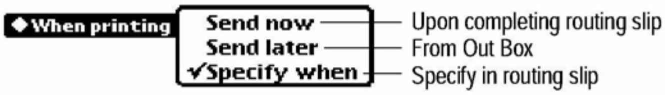
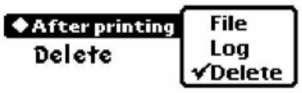
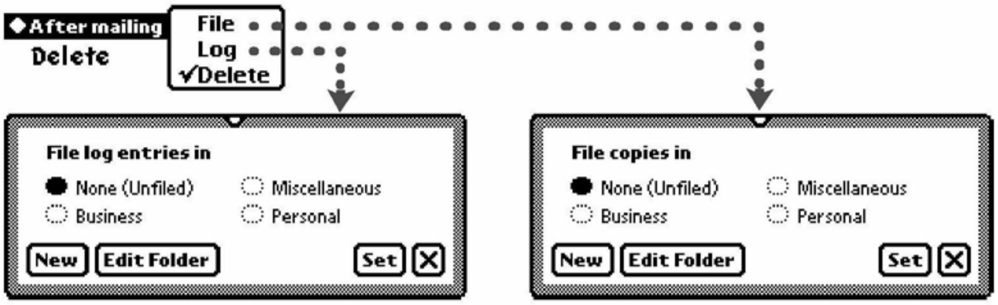
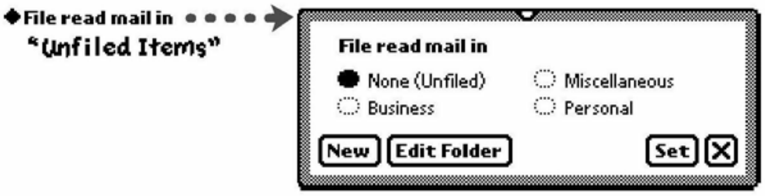
◆When faxing Specify when

◆After sending a fax
Delete

◆File read faxes in
"Unfiled Items"

Each transport may add its own preferences slip for configuring any options that apply to that transport. Figure 7-23 illustrates some common options.

Figure 7-23 Some common preference items for transports

Preference item	Example
Whether to show or hide status slips	☑ Show status dialogs
When to send	 ◆When printing Send now Send later ✓Specify when Upon completing routing slip From Out Box Specify in routing slip
What to do with sent items	
Where to file sent items or log entries	 ◆After mailing Delete File Log ✓Delete File log entries in ● None (Unfiled) ○ Business ○ Miscellaneous ○ Personal New Edit Folder Set ☒ File copies in ● None (Unfiled) ○ Business ○ Miscellaneous ○ Personal New Edit Folder Set ☒
Where to file read items	 ◆File read mail in "Unfiled Items" File read mail in ● None (Unfiled) ○ Business ○ Miscellaneous ○ Personal New Edit Folder Set ☒

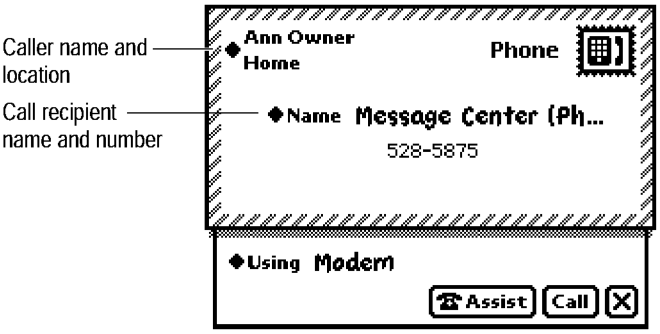
A transport's preferences slip can include other items, such as buttons. For example, the fax preferences slip includes a button for scheduling automatic fax sending times and another button for setting the Newton to receive faxes exclusively.

A preferences slip can also include an Info button in the lower left corner. Tapping it pops up an Info picker that lists at least the one item Help. Generally, picking Help from this Info picker simply displays the system help book, open to the routing section. A transport can add more items to the Info picker that pops up in a Preferences slip.

Routing Alternatives

There are some alternatives to the Action picker method of routing. For example, the Calls application has a Place Call button for routing an outgoing phone call. Tapping the Place Call button displays a routing slip for the call transport. The Call routing slip is similar to a Fax routing slip. In the upper part of the Call routing slip a user specifies the caller, the caller's location, and the call recipient (see "Sender Picker" on page 7-13 and "Choosing Fax or E-mail Recipients" on page 7-16). In the lower part of the Call routing slip, a picker and a button control how the call is dialed. Figure 7-24 shows a sample Call routing slip.

Figure 7-24 A Call routing slip sets up an outgoing phone call



Caller name and location

Call recipient name and number

◆Ann Owner
Home

Phone

◆Name Message Center (Ph...
528-5875

◆Using Modem

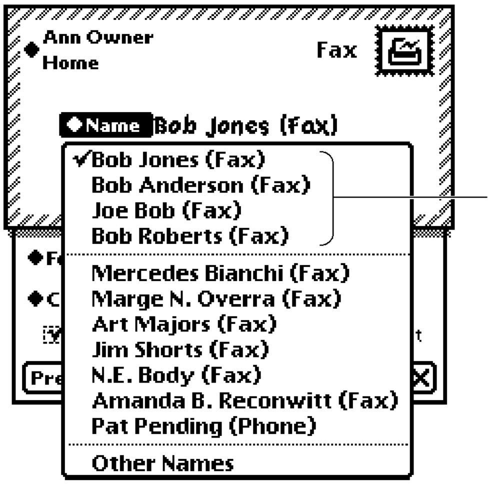
Another way users can route items through most transports is with the Intelligent Assistant. In addition, applications can route items programmatically. These two routing methods are described in more detail in the remainder of this section.

Routing by Intelligent Assistant

In addition to using an Action button to send outgoing items, a user can send items by using the Intelligent Assistant. First the user writes the name of an action—call, fax, mail, or print—and taps the Do button in the Intelligent Assistant's main view. After interpreting what action to take, the Intelligent Assistant finds out from the frontmost application which items to send. Then the Intelligent Assistant has the system display the routing slip for the type of action the user wants.

The Intelligent Assistant also interacts with the list picker for picking a recipient in a routing slip. If a user writes a routing action such as "fax Bob," the Intelligent Assistant sets up the picker with a list of names that contain "Bob" and have fax numbers from the Names File. Figure 7-25 shows how this might look.

Figure 7-25 Routing with the Intelligent Assistant



Routing slip for the action witten in the Intelligent Assistant

Names that match the recipient written in the Intelligent Assistant

Programmed Sending

An application can send an item programmatically, using a specific transport, without any user intervention. (The Action button is not used in this case.) For example, an application might have a transport make a connection whenever a user opens the application, and break the connection when the user closes the application. Another application might poll for data, such as pager messages, and could have a transport poll more frequently (and use more power) while the application is open than when the application is closed.

If your application has its own method for routing apart from the Action button, it can display a routing slip for the user to confirm or cancel the action as well as specify additional routing information. If your application routes items programmatically to an e-mail, fax, or call transport, you may want to allow users to choose the recipient. Your application can use the same method as the built-in routing slips (see “Choosing Fax or E-mail Recipients” on page 7-16). If you want to provide a way for users to select a different printer, your application can use the same printer-selection method as the Print routing slip (see “Choosing a Printer” on page 7-15). Before instituting a programmed routing action, you may want to allow the user to choose a format for the item being sent. Your application can get a list of formats that can handle the item. Using this list, the application could make available a picker from which the user could choose a format. You may also want to allow the user to choose a transport for the item being sent. Your application can get a list of transports that can handle specific formats. Using this list, the application could make available a picker from which the user could choose a transport.

CHAPTER 8

Newton Services

This chapter describes the user interface for Newton system services not described in other chapters. Topics include:

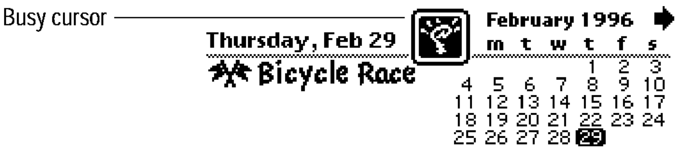
- How the system automatically indicates it is busy
- How your application or transport can allow users to hide and show status slips
- What your application should do if it schedules alarms or other actions
- What role sound should play in your application or transport
- How your application should enable users to find data in it
- How your application should enable users to file their data in folders
- How your application or transport can have the Intelligent Assistant respond to a user's written request for action
- Where and how to provide users with on-screen help
- Where and how to provide user preference settings for your application or transport

All these topics are described in terms of the applications and transports that come with most Apple MessagePad models. Applications and transports you develop should follow these models.

Automatic Busy Cursor

The system lets users know when it is temporarily busy and may be unable to respond to their input by displaying a small graphic, called the **busy cursor,** at the top of the screen. Your application or transport does not need to do anything to benefit from this feedback; the system displays the busy cursor automatically as needed. Figure 8-1 shows a busy cursor.

Figure 8-1 A busy cursor indicates the system is temporarily engaged

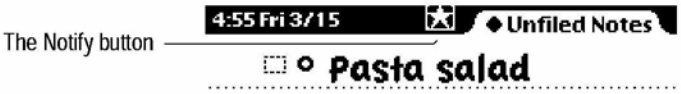


The automatic busy cursor is not meant to provide complete feedback during a lengthy operation. If your application or transport begins an operation that may take more than a few seconds to complete, it should display a status slip (see the next section and "Status Slips" on page 2-20).

Notify Button and Picker

The system displays a small graphic at the top of the screen, called the Notify button, to notify the user of ongoing actions and deferred notification alerts that applications or transports have registered with the system's Notify service. The Notify button looks like a star and it blinks periodically. Figure 8-2 shows the Notify button.

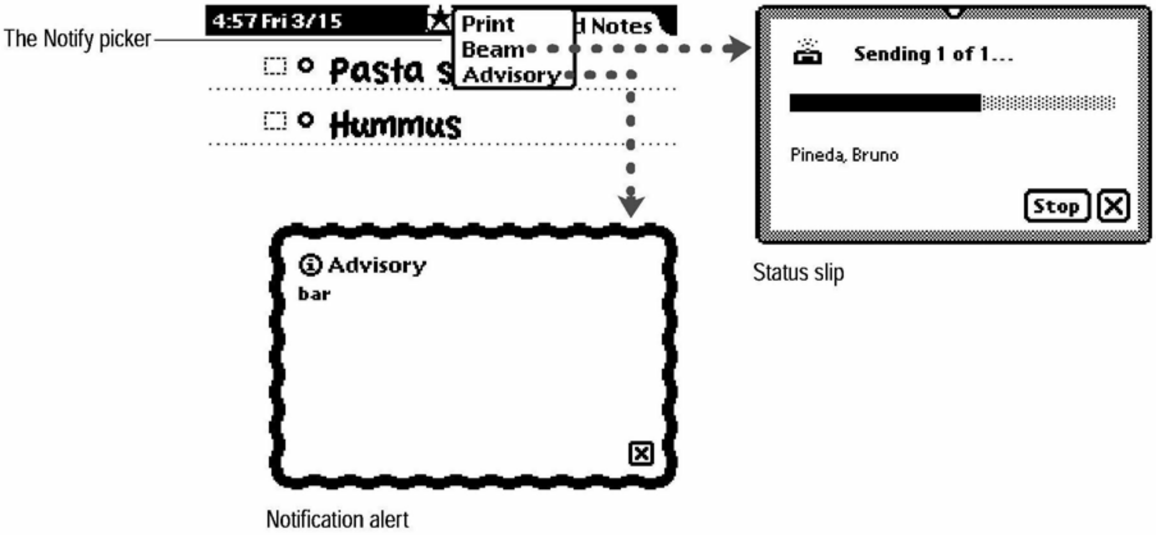
Figure 8-2 The Notify button signals an ongoing action or deferred alert



If your application displays a status slip with a Close box while it performs a lengthy action, and a user taps the Close box, your application should register the ongoing action with the Notify service. The system will continue processing the action in the background. In addition, your application may register a deferred notification alert when it sets an alarm (see “Alarms” on page 8-4) and at other times.

The Notify service lists registered actions and notifications in a Notify picker, which pops up if a user taps the Notify button. Choosing a listed action redisplays the corresponding status slip. Choosing a listed notification alert displays it. Figure 8-3 shows a Notify picker.

Figure 8-3 The Notify picker lists ongoing actions and deferred alerts



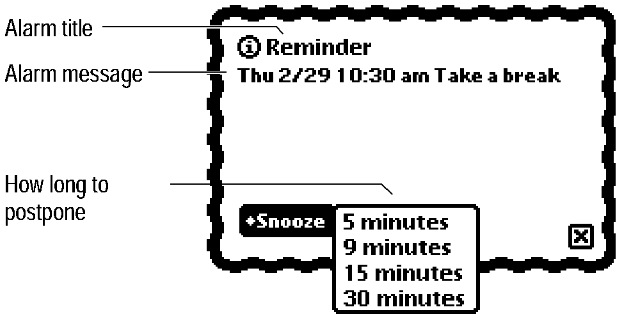
When a user chooses an action or alert from the Notify picker, the Notify service automatically removes the chosen item from the picker. If your application or transport completes an action listed in the Notify picker, it must remove the action from the Notify picker by unregistering the action with the Notify service. The Notify service automatically removes the Notify button when the last item is removed from the Notify picker.

For more information on status slips, see "Status Slips" on page 2-20 and "Routing Status" on page 7-29. For more information on notification alerts, see "Notification Alerts" on page 2-17.

Alarms

Your application can use the Newton system's Alarms service to display a notification alert or perform other actions at a specified time. If the Newton is asleep at the time the alarm is to execute, the system powers up the Newton and executes the alarm. If the alarm displays an alarm notification alert, the user can postpone the alarm for a specified time period by tapping the Snooze button included in the notification alert, as shown in Figure 8-4.

Figure 8-4 An alarm notification alert's Snooze button can postpone the alarm



ⓘ Reminder

Thu 2/29 10:30 am Take a break

The Snooze button is optional. Your application can use a plain notification alert without a Snooze button (see "Notification Alerts" on page 2-17) or no notification alert at all.

Unacknowledged Alarms

Your application does not have to do anything to handle alarms that a user does not acknowledge. If a user is away from his or her Newton when an alarm goes off and the Newton goes back to sleep before the user returns, the user won't know the alarm went off until powering up the Newton again. The Newton system's Persistent Alarms user preference solves this problem. When this preference is enabled, the system reschedules alarms until the user acknowledges them. To conserve battery power, the length of time between reappearances of a particular alarm increases in proportion to the number of times it has gone unacknowledged.

Alarm Etiquette

Every application should schedule and use alarms in a way that does not hamper the activities of other applications on the same Newton. Storage is one concern, since each alarm uses internal storage space. You don't have to limit your application to one alarm, but scheduling a daily wake-up alarm for the next year by creating 365 different alarms would use up an excessive amount of internal storage. Exercise reasonable judgment when creating multiple alarms.

Actions taken besides displaying a notification alert should be brief. If your alarm initiates a time-consuming process, it may delay the execution of other alarms set to go off at approximately the same time.

Note that your application may not be open when its alarm executes. In fact, your application may not even be installed. If your application's alarms aren't useful when it isn't installed, it should remove them when a user removes it (for example by removing the card it's on). There is no point in wasting space with useless alarms that display notification messages such as "Sorry, this alarm can't execute because the application isn't installed."

A user sets the volume of alarm sounds in the Alarm section of the Prefs application. Your application should not change the alarm volume set by the user.

Sound

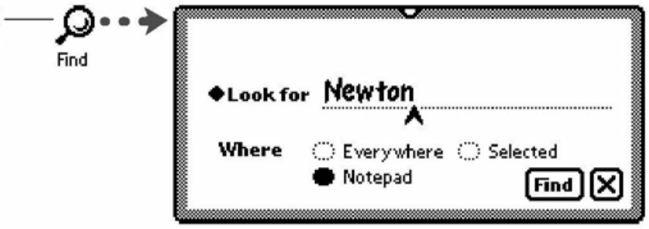
Your application can easily associate a sound with a system event or play sound on demand. Each sound can be played synchronously, so that other tasks must wait for it to finish, or asynchronously, so that another task can begin before the sound finishes. The Newton system includes a number of built-in sounds, which the built-in applications use and other applications can use as well. Newtons play only sampled (recorded or prefabricated) sounds; the Newton system does not synthesize sound.

The built-in applications and transports use sounds extensively but unobtrusively to provide secondary feedback and enrich the user experience. Sound should always play a secondary role in Newton software. Don't make sound the sole conveyer of information. For users who must turn the sound off or who are hearing-impaired, your application should convey vital information visually as well.

Find

The Newton system provides a Find service with which a user can search participating applications for text, dates, or other types of data. A user specifies what to find in a Find slip that may be supplied by the system, customized by an application developer, or supplied entirely by the developer. The Find slip appears when a user taps the Find button, which is a picture button that looks like a magnifying glass. Figure 8-5 illustrates the Find button and the Find slip that the system supplies.

Figure 8-5 A standard Find slip specifies what to find and where to look



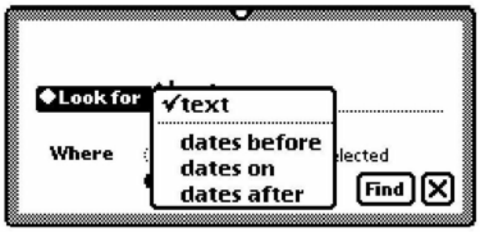
Tapping the Find button brings up the Find slip

◆Look for Newton

Where ○ Everywhere ○ Selected ● Notepad

The standard Find slip contains a labeled input line used to specify a search criterion and several radio buttons used to specify the scope of the search. The labeled input line has a picker for specifying the kind of search to perform—a text item or a date. Your application can also implement specialized searches for other kinds of data; these searches are described in more detail later in this section. Figure 8-6 illustrates the standard Look For picker.

Figure 8-6 Specifying text or date searches in a Find slip



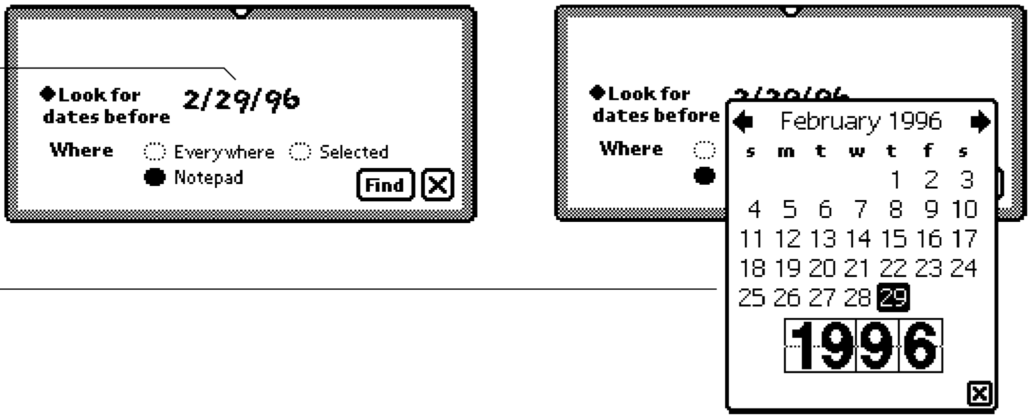
Text Searches

Text searches are not sensitive to capitalization and only match the beginnings of words. For example, a search for "man" would find the items "man" and "Manila," but not "human." To specify the text to search for, a user writes on the input line in the Find slip.

Date Searches

Date searches find items dated before, after, or on the date specified in the Find slip. To specify a date, a user taps the date shown in the Find slip. This pops up a standard Date picker, as shown in Figure 8-7.

Figure 8-7 Specifying a date in a Find slip



1. User taps the date to pop up a Date picker

2. Then selects a date in the Date picker

◆Look for dates before 2/29/96

Where ○ Everywhere ○ Selected
● Notepad

February 1996

s	m	t	w	t	f	s
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

1996

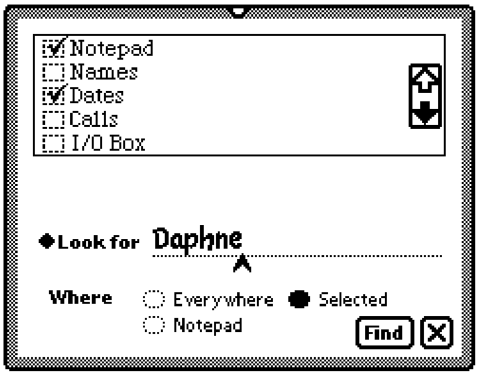
The Scope of a Search

The system-supplied Find slip always contains radio buttons labeled Everywhere and Selected. If the currently active application supports the Find service, it is represented by a radio button in this slip as well.

To search just the currently active application, a user selects its radio button in the Find slip. To search all available applications registered with the Find service, a user selects the Everywhere radio button in the Find slip. In this case, applications need not be open to be searched.

To search some applications and not others, a user taps the Selected radio button in the Find slip. Tapping the Selected radio button causes a checklist to appear at the top of the Find slip. Included in the list are all currently available applications that are registered with the Find service. The user can tap items in the list to place a check mark next to those applications in which the system is to conduct a search. The Find slip in Figure 8-8 depicts a search for the word "Daphne" in the Notepad and Dates applications.

Figure 8-8 Searching specified applications



☑ Notepad
□ Names
☑ Dates
□ Calls
□ I/O Box

◆Look for Daphne

Where ○ Everywhere ● Selected
○ Notepad

Normally the Find service searches applications in their entirety, but the currently active application can separate its data and list the separate parts in the checklist that appears when a user selects the Selected radio button in the Find slip. For example, a personal finance application could allow users to search its check register, credit card register, and accounts list independently.

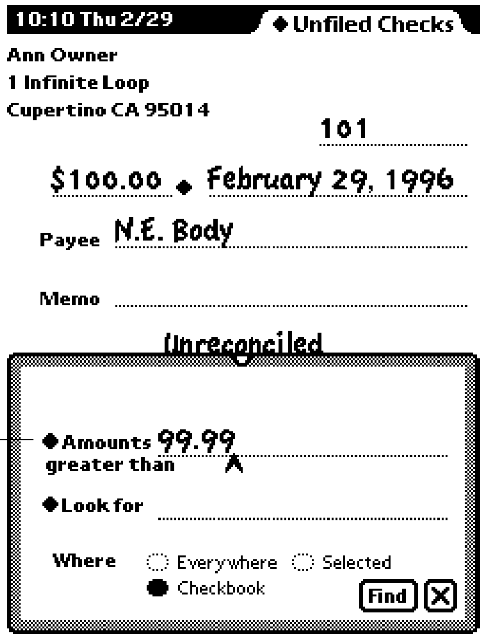
Customizing the Standard Find Slip

In addition to the system-supplied variations on the Find slip, your application can modify or completely replace the Find slip when it is frontmost. Typically, your application would do this in order to provide a customized user interface for specialized searches. For example, your application could add a labeled input line with a picker that enables a user to conduct specialized finds.

If your application specifies custom interface elements, the system adds them to the top portion of the standard Find slip whenever two conditions are met. First, your application must be frontmost. Second, a user must select your application's radio button in the Find slip.

Keep in mind that a user may need to scroll among found items while the Find slip is displayed; therefore, when customizing or replacing this slip, avoid making it so large that it obscures the display of the found items. Figure 8-9 shows a sample application named Checkbook that adds a labeled input line with a picker to the standard Find slip.

Figure 8-9 A custom Find slip displays application-specific criteria at the top



10:10 Thu 2/29 ◆Unfiled Checks

Ann Owner
1 Infinite Loop
Cupertino CA 95014

101

$100.00 ◆ February 29, 1996

Payee N.E. Body

Memo

Unreconciled

◆Amounts greater than 99.99

◆Look for

Where ○ Everywhere ○ Selected ● Checkbook

Criterion added by the frontmost application

Besides adding criteria to the top of the Find slip, your application can suppress the Find slip's standard input line. If necessary, your application can completely replace the standard Find slip with a slip of its own.

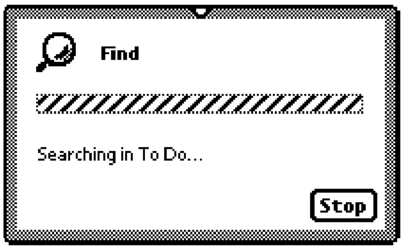
Initiating or Canceling a Search

After using the Find slip to specify the search criteria, a user initiates the search by tapping the Find button. Alternatively, the user can cancel the search by tapping the Close box to dismiss the Find slip.

Search Status

While a search executes, the system reports its progress in a status slip. Figure 8-10 depicts a typical Find status slip.

Figure 8-10 A status slip shows the progress of a Find operation



Find

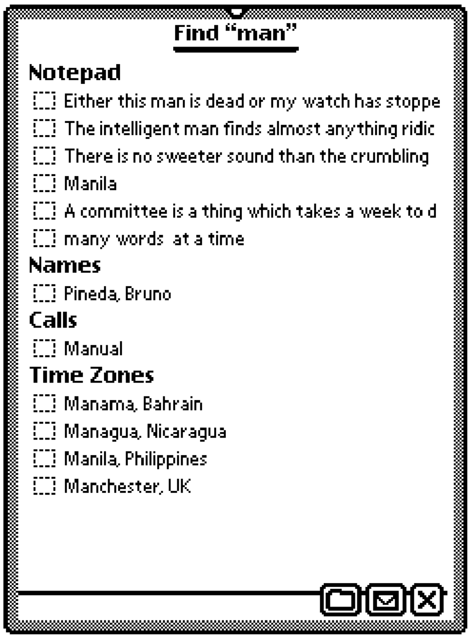
Searching in To Do...

The status slip displayed by the Find service includes the Find icon and name, an animated busy indicator that looks like a barber pole, and the name of the application currently being searched. The status slip also includes a Stop button, which allows the user to halt the search in progress.

Search Results

If a search finds just one item, that item is displayed behind the Find slip. If a search finds more than one item, the Find service displays an overview list of the found items. Figure 8-11 depicts a Find overview as it might appear after searching all applications for "man."

Figure 8-11 A Find overview lists items that match search criteria



Find "man"

Notepad

□ Either this man is dead or my watch has stoppe
□ The intelligent man finds almost anything ridic
□ There is no sweeter sound than the crumbling
□ Manila
□ A committee is a thing which takes a week to d
□ many words at a time

Names

□ Pineda, Bruno

Calls

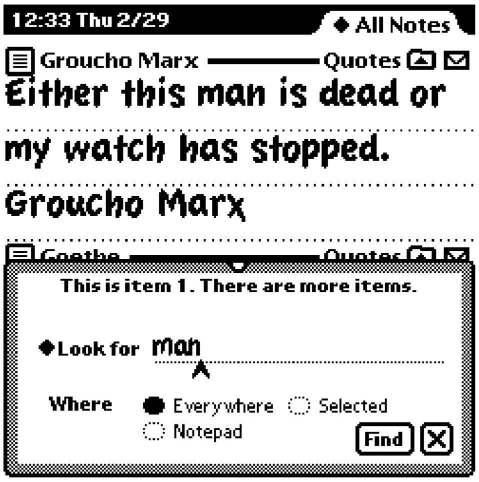
□ Manual

Time Zones

□ Manama, Bahrain
□ Managua, Nicaragua
□ Manila, Philippines
□ Manchester, UK

A user can alternately hide and reveal the names of items listed under an application in the Find overview by tapping the application name there. Tapping the name of an item in the Find overview displays a detail view of the item. The Find slip stays open in front of the detail view. A message at the top of the Find slip states which item is displayed and whether other items were found in the same application. Figure 8-12 shows an example of the message.

Figure 8-12 The Find slip reports which found item is currently displayed



Either this man is dead or my watch has stopped.
Groucho Marx

This is item 1. There are more items.

◆Look for man

Where ● Everywhere ○ Selected ○ Notepad

If more than one item was found, tapping the universal down arrow goes to the next found item, and tapping the universal up arrow goes to the previous found item. Tapping the Overview button redisplays the overview of found items.

Between uses, the Find service stores the setting of the Look For picker. The next time a user taps the Find button, the Find slip reopens with the most recent search criteria preset. Note that in order to conserve memory, the overview list of found items is not saved between uses of the Find service.

Filing

The Newton system's Filing service allows users to associate data items with folders displayed by the user interface. A user can create, edit, and delete folders at will. In addition, users can select a storage location—internal or card—with the Filing service.

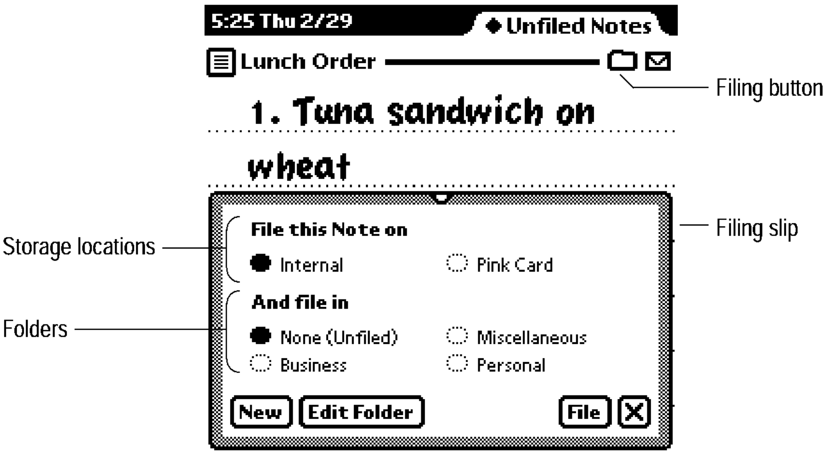
Filed data items look to a user like they are in folders, but filed items do not actually reside in a folder or directory structure. Instead, the Filing service tags a filed item to identify the folder in which it belongs. When a user wishes to see an application's data items belonging to a particular folder, the application displays the data items having the appropriate tag.

Applications use two user interface elements to implement filing: a Filing button and a folder tab. The Filing button enables users to file application data in folders. After filing items in folders, users can locate them by using the folder tab and its picker. (For more information on the folder tab, see page 8-19.)

Filing Button and Slip

If you want users to be able to file data in your application, it must include a Filing button, which is a picture button that looks like a an ordinary file folder. Tapping a Filing button displays a Filing slip, which enumerates the available filing options. Figure 8-13 shows a sample Filing button and Filing slip.

Figure 8-13 A Filing slip names available folders and storage locations



5:25 Thu 2/29

◆ Unfiled Notes

Lunch Order

1. Tuna sandwich on wheat

File this Note on

● Internal ○ Pink Card

And file in

● None (Unfiled) ○ Miscellaneous
○ Business ○ Personal

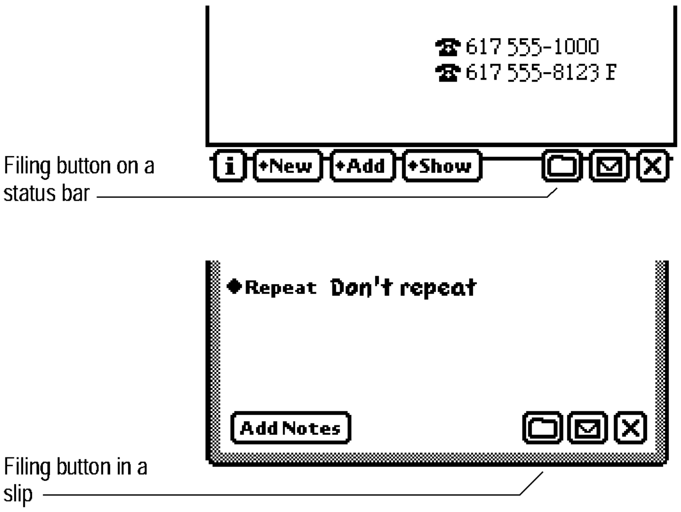
The filing options that a user selects in a Filing slip apply to the data in the view that contains the Filing button. If the view contains multiple data items that can be individually selected, such as the items listed in an overview, then the filing options selected by the user apply to the currently selected items.

If there is nothing to file when a user taps a Filing button—for example, the user taps the Filing button in an overview without first selecting any items—the Filing service displays the message "There is nothing to file."

A Filing Button's Location

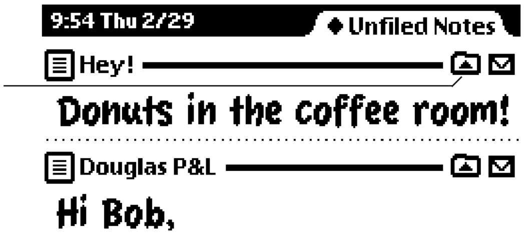
Where you put a Filing button in your application's views depends on how much data the Filing button affects. If a Filing button affects all the data in a view, then it should go at the bottom right corner of the view, next to the view's Action button. For example, the main view of the Names File application has one Filing button, and it affects all data in the view. Figure 8-14 shows examples of views with one Filing button each.

Figure 8-14 A Filing button at the bottom of a view affects the entire view



In a view where a Filing button can affect only one data item of several that may be displayed in the view (perhaps by scrolling the view), there should be a Filing button above each item, at the right side of the view. Generally such a view has a separator bar above each data item, and a Filing button should be at the right end of each separator bar. For example, each note in the Notepad application has its own Filing button, which applies just to that note. Figure 8-15 shows an example of Filing buttons above each data item in a view.

Figure 8-15 A Filing button above an item affects only that item



Donuts in the coffee room!

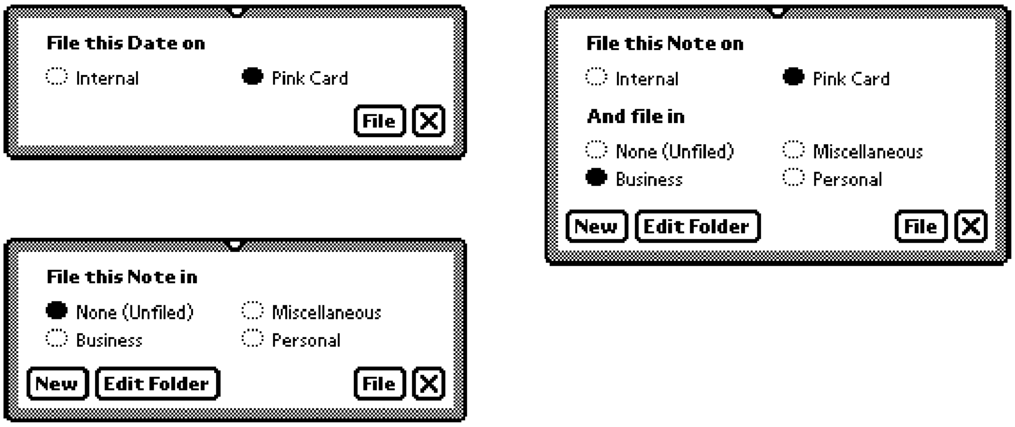
Hi Bob,

Filing button on a separator bar

A Filing Slip's Contents

A Filing slip contains one or two clusters of radio buttons, one for selecting a storage location and one for selecting a folder. An application can suppress either cluster. For example, the built-in Date Book application suppresses the cluster for selecting a folder. Furthermore, the system suppresses the cluster for selecting a storage location if the Newton has only one storage location. However, the Filing slip always contains at least one cluster of radio buttons. If your application suppresses the folders cluster, the system does not suppress the storage-locations cluster even if there is only one storage location available. Figure 8-16 illustrates several configurations of the Filing slip.

Figure 8-16 A Filing slip can include storage locations, folders, or both



File this Date on

○ Internal ● Pink Card

File this Note on

○ Internal ● Pink Card

And file in

○ None (Unfiled) ○ Miscellaneous

● Business ○ Personal

File this Note in

● None (Unfiled) ○ Miscellaneous

○ Business ○ Personal

A Filing slip should open with the current folder and storage location selected. Your application can override this behavior if it cannot determine a useful initial filing state, such as when a user has selected multiple items for filing from an overview.

A Filing slip includes folders that are visible only in the current application as well as folders that are visible everywhere. All the folders are listed together, in alphabetical order. If necessary, you can have your application suppress the display of either type of folder. For example, the built-in Extras Drawer application only displays folders specific to it. Note that this option to suppress folders should not be a user preference, but should be decided when the application is designed.

Applications provide the headings for the radio button clusters in a Filing slip. The wording your application should use for headings depends on the how many items a user is filing and on whether the Filing slip includes both folders and storage locations or just one of them. Table 8-1 specifies the wording to use in each case.

Table 8-1 Headings for radio button clusters in Filing slips

Heading for storage locations	Heading for folders	Number of items being filed
File this Item on	—	one
File this Item on	And file in	one
File the selected Items on	—	multiple
File the selected Items on	And file in	multiple
—	File this Item in	one
—	File the selected Items in	multiple

NOTE In these headings the word Item is capitalized and may be replaced with the specific type of item being filed, such as Note, Name, Date, Task, Call, and so on. None of these headings ends with a colon or has any other punctuation.

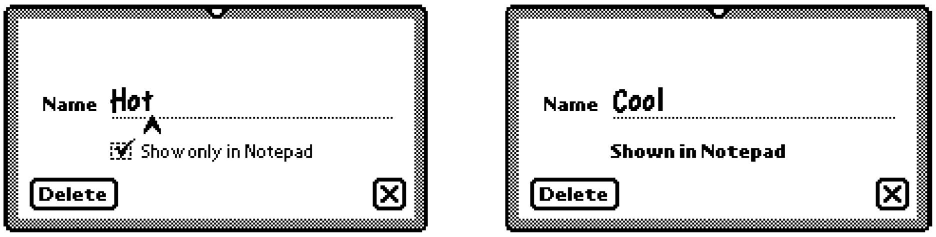
In addition to radio buttons for selecting filing options, all Filing slips have a File button for initiating the filing operation and a large Close box for canceling the filing operation.

Editing Folders

If a Filing slip contains radio buttons for selecting a folder, the slip also includes New and Edit Folder buttons for creating new folders and editing the names of existing ones. Tapping either button displays a slip for entering and editing a folder name. Figure 8-17 shows examples of the slips used for entering and editing folder names.

In a slip for creating a new folder, the slip includes a checkbox for designating where the folder can be seen—in all applications (everywhere) or just in the application where the folder was created. In a slip for editing an existing folder, the checkbox is replaced by a message stating where the folder name is shown. The same message appears instead of a checkbox when creating a new folder in an application that suppresses the display of folders specific to it or the display of folders visible everywhere (as described earlier in “A Filing Slip’s Contents” on page 8-16).

Figure 8-17 Slips for entering and editing folder names



Name Hot

☑ Show only in Notepad

Delete ☒

Name Cool

Shown in Notepad

Delete ☒

Folder Creation slip has a checkbox for designating where the folder can be seen

Folder Editing slip reports where the folder can be seen

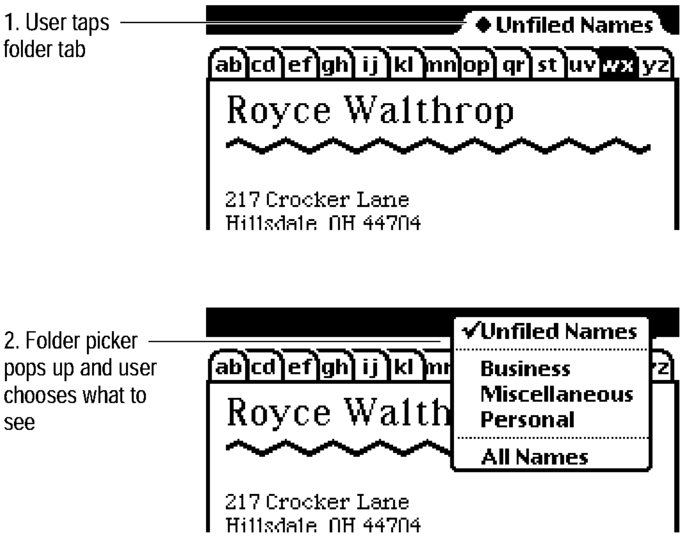
Users can create up to 12 folders visible everywhere and 12 more folders specific to each application. The system does not permit an application-specific folder to have the same name as a folder that is visible everywhere.

Folder Tab

If a view has a Filing button (for filing the view's data into folders), then the view must also have a folder tab at the top so users can see the data they have filed in folders. A folder tab looks like the cut tab part of a paper file folder. The Newton folder tab shows the name of the folder whose data is currently displayed in the view. The name begins with a diamond because tapping it pops up a picker from which a user can choose which folder to see. Additionally, a user can choose to see only items not filed in any folder or items from all folders (including unfiled items). A check mark appears next to the current choice. Figure 8-18 shows a sample folder tab and picker.

Every Folder picker includes two choices in addition to the alphabetical list of folders. At the top of the Folder picker, above a separator line, is the choice Unfiled Items. At the bottom of the alphabetical list of folders, below a separator line, is the choice All Items. In both of these choices, your application can replace the word Items with the name of the type of item displayed in the view.

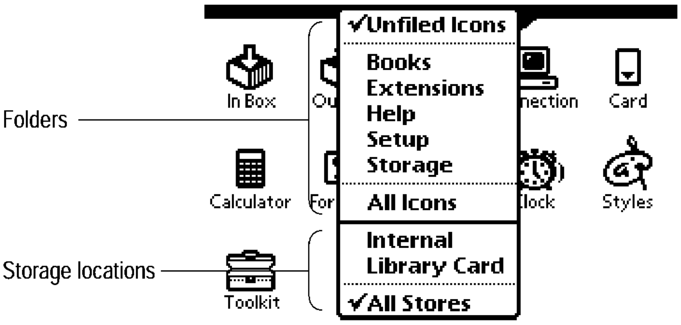
Figure 8-18 A folder tab allows users to filter a view by folder


1. User taps folder tab

2. Folder picker pops up and user chooses what to see

At the bottom of a Folder picker, below a solid separator line, your application can have the system list available storage locations. This allows a user to specify a storage location in addition to a folder from which to display items. The system does not list storage locations in a Folder picker unless the Newton has more than one storage location. Figure 8-19 shows an example of a Folder picker with storage locations listed at the bottom.

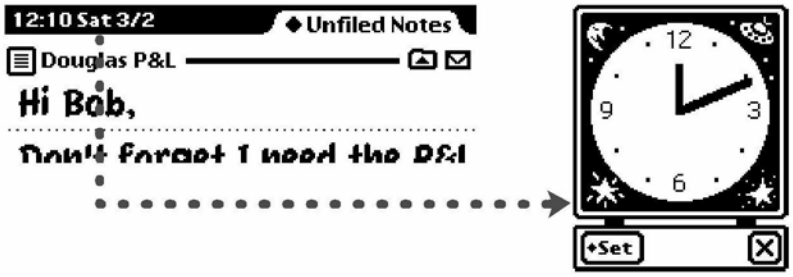
Figure 8-19 A Folder picker can list available storage locations


Folders

Storage locations

A variation on the plain folder tab includes a digital clock and calendar that a user can tap to display the built-in Clock application. Figure 8-20 illustrates a folder tab with clock calendar.

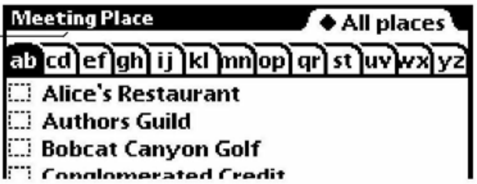
Figure 8-20 A folder tab can include a digital clock and calendar



Tapping the date and time displays the Clock application

Instead of a digital clock and calendar, your application can display a view title in the black area of a folder tab, as shown in Figure 8-21.

Figure 8-21 A folder tab can include a view title



□ Alice's Restaurant
□ Authors Guild
□ Bobcat Canyon Golf
□ Conglomerated Credit

View title

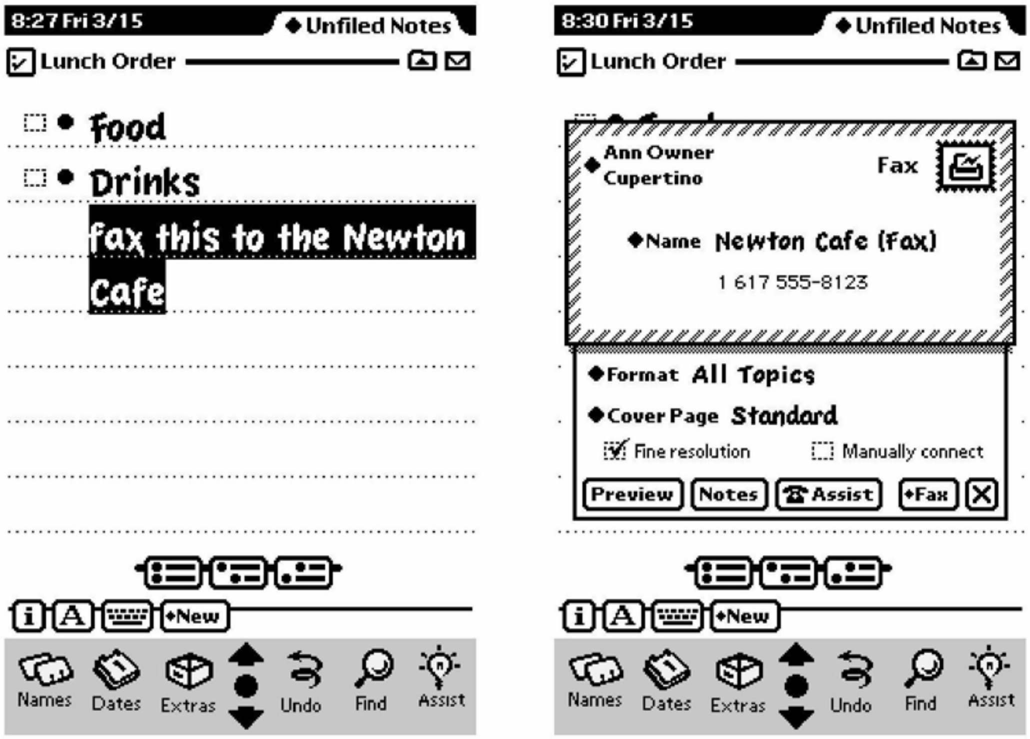
Intelligent Assistant

The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly.

Invoking the Assistant

A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Figure 8-22 The Assist button makes the Assistant try a written action request



8:27 Fri 3/15 ◆Unfiled Notes

☑ Lunch Order

□ • food

□ • Drinks

fax this to the Newton Cafe

8:30 Fri 3/15 ◆Unfiled Notes

☑ Lunch Order

◆Ann Owner Cupertino

Fax

◆Name Newton Cafe (Fax)

1 617 555-8123

◆Format All Topics

◆Cover Page Standard

☑ Fine resolution □ Manually connect

1. User selects written action request and taps Assist button

2. Assistant attempts to complete the requested action

Interpreting the Request Phrase

The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same task if given the word “phone,” “call,” or “dial.” Applications can add words to the Assistant’s lexicon, but users cannot.

The Assistant matches words regardless of their capitalization. For example, it considers the word “phone” to be the same as the word “Phone.”

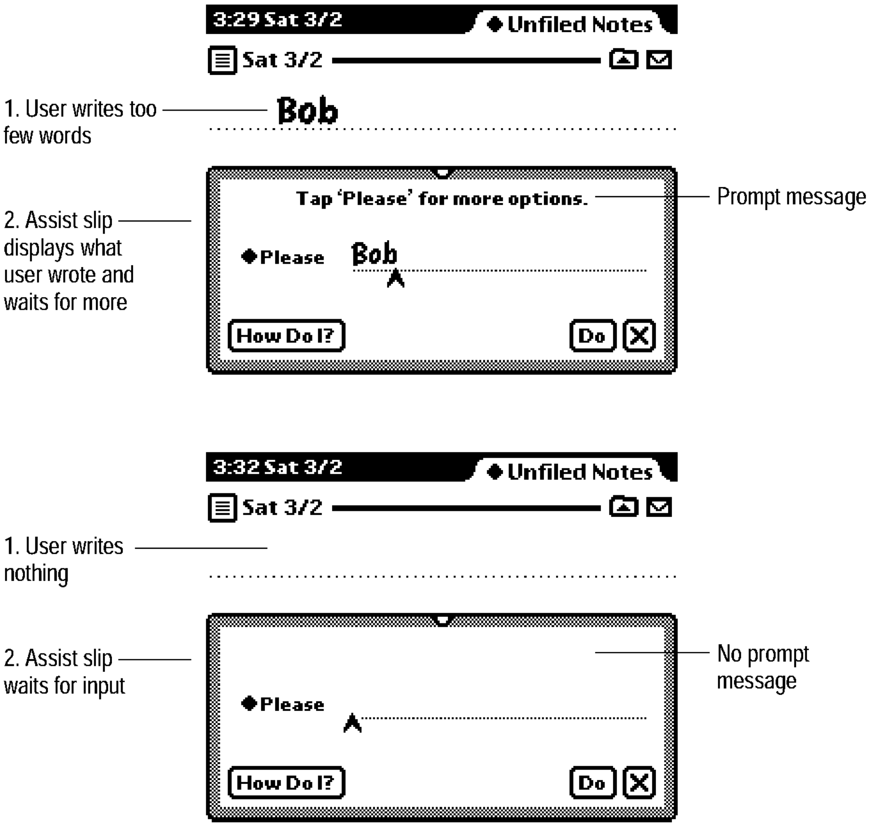
The order in which a user writes words is not significant. For example, the phrase “Royce fax” produces the same result as the phrase “fax Royce.” This syntax-independent architecture allows easier localization of applications for international audiences.

The Assistant ignores words it does not know, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as “Make a phone call to Bob at work” and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words.

Assist Slip

If the Assistant can’t interpret a user’s written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip’s Please picker and can write on the slip’s input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote “Bob,” the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words.

Figure 8-23 An Assist slip appears when the Assistant needs more information



Tap 'Please' for more options.

1. User writes too few words

2. Assist slip displays what user wrote and waits for more

Prompt message

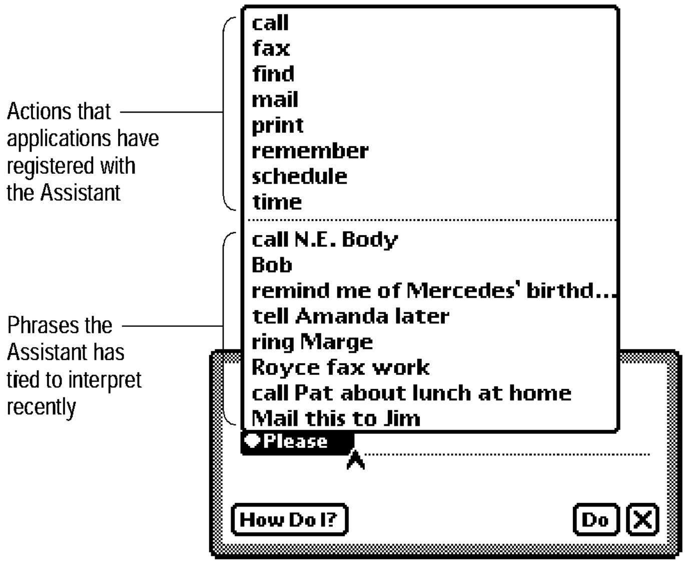
1. User writes nothing

2. Assist slip waits for input

No prompt message

An Assist slip's Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker.

Figure 8-24 The Assistant's Please picker lists known actions and recent phrases


Actions that applications have registered with the Assistant
Phrases the Assistant has tied to interpret recently
call
fax
find
mail
print
remember
schedule
time
call N.E. Body
Bob
remind me of Mercedes' birthd…
tell Amanda later
ring Marge
Royce fax work
call Pat about lunch at home
Mail this to Jim
Please
How Do I?
Do
☒

The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your application registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top portion of the Please picker. For example, the word "call" appears but the synonyms "ring" and "phone" do not. Recently used synonyms may appear in the bottom half of the picker, however.

If a user writes a verb the Assistant doesn't know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase "buzz 555-1234" the Assistant does not match the word "buzz" to an action, but it can identify "555-1234" as having the format of a telephone number. Based on that information, the Assistant deduces the user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker.

In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see "Help" on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action.

The primary function of an Assist slip is to specify an action. If the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip.

Task Slips

Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip.

For example, if a user writes the request "fax Bob," the Assistant can get Bob's fax number from the Names File application. But what if Bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page.

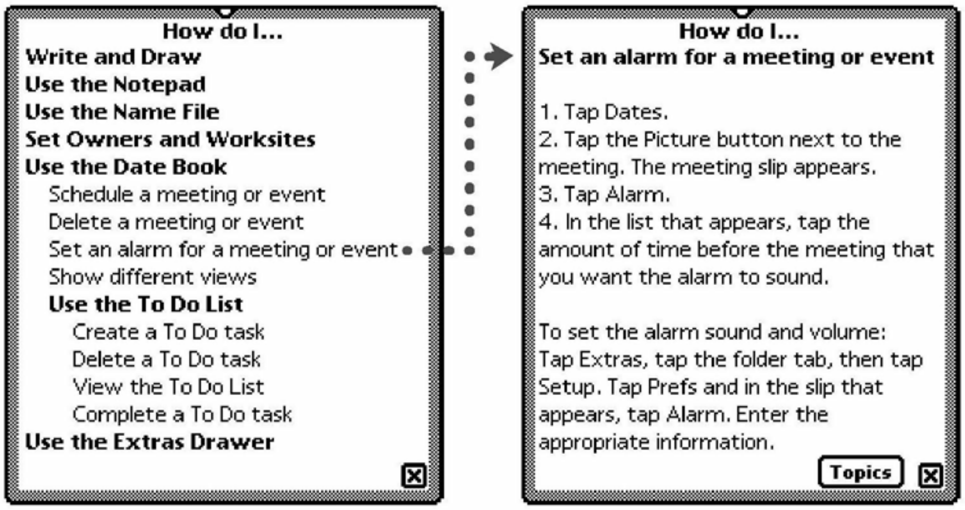
The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions—printing, faxing, mailing, or calling—the task slip is a routing slip (see "Routing Slips" on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see "Find" on page 8-6).

Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way.

Help

The Newton system includes online help for built-in applications and system services. Users can see the online system help by tapping the How Do I? button in the Assist slip. When a user taps that button, the system displays an outline-like overview of help topics. Tapping a topic expands the outline to display that topic's subtopics or, if that topic has no subtopics, to display instructions for that topic. Although you cannot add to the built-in help, you can provide the same kind of help within your Newton application. Figure 8-25 shows the built-in help overview.

Figure 8-25 Online help has a topical outline and concise instructions



Set an alarm for a meeting or event

1. Tap Dates.
2. Tap the Picture button next to the meeting. The meeting slip appears.
3. Tap Alarm.
4. In the list that appears, tap the amount of time before the meeting that you want the alarm to sound.

To set the alarm sound and volume: Tap Extras, tap the folder tab, then tap Setup. Tap Prefs and in the slip that appears, tap Alarm. Enter the appropriate information.

Users can also access the built-in help by choosing Help from the Info picker in any built-in application. When accessed through an Info picker, the help overview appears with the appropriate outline topic already expanded. Likewise, your application gives users access to its online help through its Info picker (see "Info Picker" on page 4-24).

Another method of accessing online help is through the Extras Drawer. The built-in help is a digital help book that is customarily filed in the Help folder of the Extras Drawer. A user can open the built-in online help by tapping the Help icon there. You can give users the same access to your application's online help by making it a digital help book in the Extras Drawer. For more information on digital help books, see *Newton Book Maker User's Guide.*

The purpose of online help is to provide quick access to single screens of step-by-step instructions for performing actions in a Newton application. If you add online help to your application, keep the following points in mind:

- Organize your online help so users can't get lost in it.
- Make help information short, since the help view doesn't scroll and probably never will. The system truncates help information that exceeds the length of the help view.
- Keep each help page simple, specific, and task-oriented.
- Phrase each overview topic and subtopic so it grammatically completes a question that begins "How do I?" For example, the topic "Use the Shopping List Application" asks the question "How Do I Use the Shopping List Application?" Under this topic could be subtopics such as "Add Items to the Shopping List" and "Check Off Purchased Items." Do not begin a topic or subtopic with a gerund, such as "Using," or the topic will not mesh grammatically with the "How do I?" question.

Online help is not intended to provide a full user manual. If you want to create an online user manual in a large view with multiple-font text and on-screen controls for content navigation, make it a regular Newton digital book (not a digital help book). For information on making digital books, see *Newton Book Maker User's Guide.*

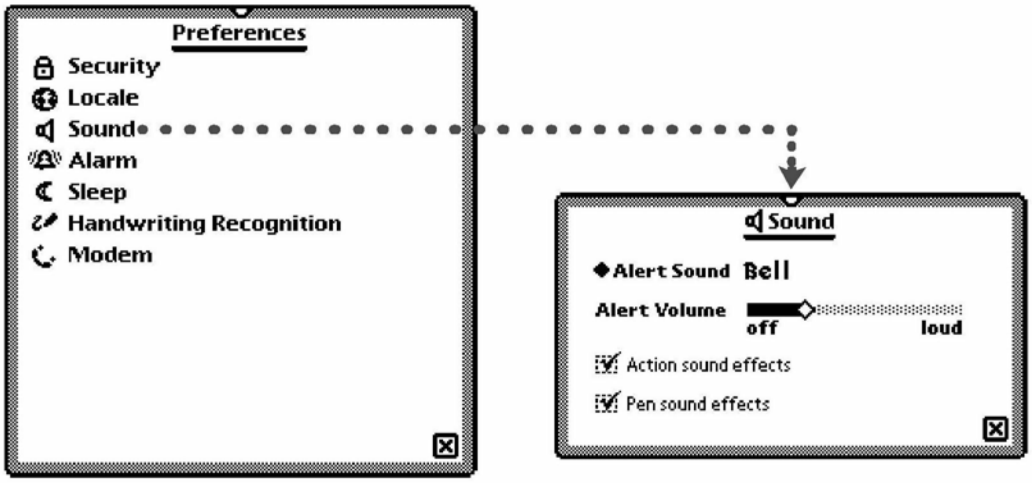
Preferences

Users can see and change two types of preference settings: system-wide and application-specific.

System-wide Preferences

A user accesses system-wide preferences through the built-in Prefs application. Its main view lists preference categories, and tapping a category displays a slip containing relevant preference items. Your application can add categories and corresponding views that the Prefs application displays, but they must be for system-wide preferences rather than application-specific ones. Figure 8-26 shows the standard system-wide categories on a MessagePad 120.

Figure 8-26 The Prefs application shows system-wide preference settings



Alert Sound Bell

Alert Volume off loud

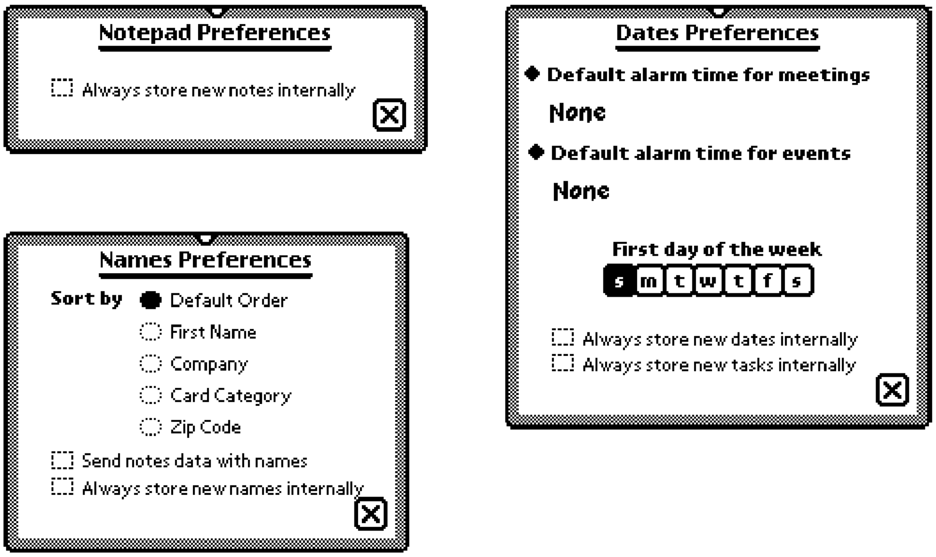
☑ Action sound effects

☑ Pen sound effects

Application Preferences

Applications provide access to their preference settings through the Info picker (see “Info Picker” on page 4-24). When a user chooses Prefs from an Info picker, the application displays a preferences slip in which the user can see and change application-specific preference settings. Figure 8-27 shows the preferences slips for some built-in applications.

Figure 8-27 A preferences slip contains application-specific settings



Notepad Preferences

□ Always store new notes internally

☒

Names Preferences

Sort by ● Default Order
○ First Name
○ Company
○ Card Category
○ Zip Code

□ Send notes data with names
□ Always store new names internally

☒

Dates Preferences

◆ Default alarm time for meetings

None

◆ Default alarm time for events

None

First day of the week

s m t w t f s

□ Always store new dates internally
□ Always store new tasks internally

☒

If your application stores data items, its Preferences slip should aid users in managing data by including a checkbox for setting the storage location of new items. This checkbox should have a label similar to “Always store new items internally,” where the word *items* is replaced by the application’s particular type of item. This check box helps to mitigate the problem users commonly have of managing where their data is stored.

Preferences should be settings that users change infrequently. If you provide choices to users that they will change many times while working with your application, you should implement those choices with a button and picker on the status bar or some other interface element to which users have easy access.

APPENDIX

Avoiding Common Mistakes

This appendix summarizes what you should do to avoid the top 20 user interface mistakes.

Info Button

Use the Info button—with the “i” icon—and its picker for information options such as Help, About, and Prefs. Always place the Info button at the far left end of the status bar unless your application includes an Analog Clock, which is optional. See pages 3-23, 4-24, 8-28, and 8-30.

New and Show Buttons

If users can create new items or display different views of information in your application, include a New button and a Show button like the ones in the built-in applications. Put the New button near the left end of the status bar next to the Info button (if present), and put the Show button to the right of the New button. See page 3-26.

Screen Size

Design your application to handle any screen size and aspect ratio. If your application can’t scale its views small enough or can’t rearrange view contents to fit the aspect ratio, notify the user before closing your application. See pages 1-11 and 2-34.

Tapping v. Writing

Tapping is faster than writing, so for data input favor pickers, scrolling lists and tables, radio buttons, sliders, and so forth over written input. See page 6-3.

Picker Placement and Alignment

Align the top of a picker with the top of its button or label. Make exceptions for overview pickers, for other very wide or very tall pickers, or for small screens. See page 4-8.

Display a picker so its button or label is at least partially visible, and keep the button or label highlighted while the picker is open. (An overview picker can cover the label or button that makes it appear.) See pages 4-9 and 4-20.

Field Alignment

Be consistent in how you align field values with field labels (including picker labels). Generally you should line up a field's label with the field's displayed value, not with the dotted line (if present) on which a user edits the field value. In a view that has several fields in a column, line up the labels at their left edges to insure a neat, orderly appearance for your application. See page 6-2.

Close Box Size

Use a regular (small) Close box in a view where there are no adjacent buttons. Use a large Close box only where there are adjacent text buttons or standard-height picture buttons. See pages 3-15 and 3-15.

Button Location

Put buttons that affect an entire view at the bottom of the view, and put buttons that only affect part of the view elsewhere. Group buttons that affect content and appearance at the bottom left of a view, and put buttons that control or initiate action at the bottom right. See page 3-11.

Button Spacing

Space adjacent buttons three pixels apart, and leave four pixels between buttons and the border of the view they're in. See page 3-12.

Button Size

Make every text button 13 pixels high and center the button's name vertically. Make the button just wide enough that with the button's name horizontally centered there are three or four pixels between the name and the button's left and right borders. See pages 3-3 and 3-7.

Capitalization

Capitalize the following items like sentences: checkboxes, field labels, and picker items. Capitalize the following items like book titles: view titles, text button names, and radio buttons. In some contexts it makes sense to capitalize differently, but your should be consistent within an application. See pages 2-5, 3-4, 3-18, 3-19, 4-3, 4-20, and 6-2.

Picker Icons

Think twice before including icons in pickers. They're hard to design and have limited benefit. See page 5-12.

Dismissing a Slip

If dismissing a slip does not cause an action to take place (other than accepting changes made to data in the slip), use a Close box for putting away the slip. In this context the Close box means "close" or "put away." Use a take-action button and a Close box if users have a choice when dismissing the slip of initiating an action or canceling. In this context the Close box means "cancel." See pages 2-16, 2-23, and 2-33.

Take-Action Button

Name a slip's take-action button with a specific verb such as Print, Fax, or File. Only use vaguely affirmative names such as OK and Yes where you want to force users to scan other parts of the slip to verify what action the button initiates. See pages 3-4 and 3-5.

Fonts

Use fonts carefully. For the voice of the system and application use the bold style of the System font in 9- or 10-point sizes. For values a user can change use Casual 10- and 12-point. (Those are the fonts that are preset by the system protos.)

Keyboard Button

If your application includes a Keyboard button on the status bar or at the bottom of a slip, use the larger-size button (as in the Notepad) unless space on the status bar is constrained (as in the Date Book). See pages 3-25 and 6-33.

Punctuation to Avoid

Don't use ellipses (...) in button names, picker labels, or list-picker items. See pages 3-4 and 4-3.

Do put an ellipsis at the end of the title or the message text in a status slip, but use three periods rather than an ellipsis character. Also use an ellipsis to accommodate an item whose text is too long to fit on a line in the space available for it (for example, in overviews). See pages 2-21, 2-45, and 6-5.

Don't use a colon at the end of a title, a heading, or a field label. See pages 2-5, 3-18, and 6-2.

Extras Drawer Icons

To avoid overlapping icons in the Extras Drawer, make yours no more than 29 pixels tall and wide. Leaving a little space helps separate icons. See page 5-8.

Limit the length of an Extras Drawer icon's name to between 9 and 11 characters per line. Put a blank space in the name where you want it to break and wrap onto another line. See page 5-9.

Make a Newton icon more distinctive and easier to identify by giving it a distinctive silhouette rather than a boxy shape. See page 5-3.

Storage

Allow users to move your application's data between storage locations with the Filing button in the Extras Drawer's status bar. This is the method used by the built-in applications. See page 8-14.

Date and Time Input

To input dates and times use the specially designed Newton pickers. See page 4-17.

Glossary

alert box A view that appears on the screen to warn the user or report an error.

alert sound An audible warning from the Newton's speaker that warns the user of an unusual or potentially undesirable situation. An alert sound may or may not be accompanied by a notification slip.

application Software that performs a specific task, such as the Notepad, Date Book, and Names File.

application base view
The container view that contains all other views that make up an application. Compare to **main view.**

auto-repeat The repeated automatic generation of characters that happens when a user holds down the pen on an on-screen keyboard.

backdrop The one application that cannot be closed. Initially the Notepad is the backdrop, but a user can use the Extras Drawer to make a different application the backdrop.

bitmap A set of bits in the Newton's memory that represent the pixels of a picture.

busy cursor A graphical signal that the system displays automatically while it is temporarily unable to process user input.

button A small graphic object that performs an action when tapped. See also **picture button** and **text button.**

button bar A thick black line with buttons on it.

caret A symbol (^) displayed where the Newton will next insert text that a user writes, prints, or types.

character Any symbol that has a widely understood meaning and thus can convey information. Some characters—such as letters, numbers, and punctuation—can be displayed on the Newton, faxed, sent in an e-mail message, and printed on a printer.

checkbox A standard Newton control that displays a setting, either checked (on) or unchecked (off). Tapping a checkbox or its text label reverses its setting. One or more checkboxes can be checked. Compare to **radio button.**

close To make a container view go away by tapping the Close box.

Close box A small square box with an X inside, located in the lower right corner of a container view. Tapping it closes the container view. Compare to **large Close box.**

command An instruction that causes the Newton or a device connected to it to perform some action. The user issues a command by tapping a button or choosing an item from a picker.

confirmation slip A view that appears on-screen to have the user confirm or cancel an action that may have far-reaching consequences.

container view A framed object that displays information (text, graphics, or both) and may contain controls that the user operates by tapping, as well as areas where the user can write and draw.

context-sensitive Describes an application that can adjust its actions according to the current situation. For example, an application with context-sensitive user input adjusts handwriting recognition according to the type of field (name, date, time, number, phone number, and so on).

control An object in a container view that a user can manipulate with a pen to cause instant action with visible results or to change settings to modify a future action.

Date Book The built-in application for recording and viewing appointments and calendar notes, setting alarms, entering repeating events, and keeping a to-do list.

dimmed Describes words or objects that appear gray. Do not dim text or objects in Newton applications; hide objects that are disabled or unavailable.

divider bar See **separator bar.**

double-tap To touch the same spot, or nearly the same spot, twice in rapid succession with the pen.

drag To place the pen on a movable object, slide the pen to move the object, and lift the pen to stop moving the object.

drag handle A small control that a user can drag to move a container view. It is a small black tab with a central hole, and is centered in the view's top border.

drawer A container view that slides open and closed at the bottom of another container view.

edit To change or modify. For example, to insert, remove, replace, or move text.

electronic ink The marks a user sees as the user writes or draws on the screen, as opposed to the typeset words or regular shapes the system displays when it recognizes the user's writing or drawing.

endpoint A type of communications connection such as a serial connection, modem, infrared beam, or AppleTalk network.

expando An input area that expands when tapped to become large enough for writing.

Extras Drawer A built-in container view that displays named picture buttons a user can tap to open applications.

field One item of data input. Also, the place in a container view where a user can input a data item by tapping, typing, writing, or drawing.

floating container view
A container view that initially appears in front of all open **sibling views.**

folder tab A control that allows users to select which folder's contents are currently displayed in a container view. The folder tab goes at the top of the container view and displays the name of the currently selected folder.

font A complete set of characters in one typeface design.

font size The size of a font in **points.** Examples of font size are 12 point and 18 point.

font style A set of stylistic variations other than size, such as bold, italic, and underline.

gauge An object with a marker that indicates an amount, degree, or value in relation to a range of possible values. A user can only read a gauge. Compare to **slider.**

glance A small container view that closes itself automatically after it has been displayed for a brief period. Also, if a user taps the view, it closes immediately.

gravity A drawing feature that causes the endpoints of a newly drawn line to snap to nearby corners and midpoints of existing graphic shapes.

highlight To make something visually distinct, typically when it's selected. Usually done by reversing black and white areas.

hot spot A small unnamed control that responds like buttons when tapped. Usually there are many hot spots in a view, and they can be visible or not.

icon A symbol that graphically represents an object or a concept. For example, icons in the Extras Drawer represent applications.

Item Info slip A slip that reports statistics for an item headed by a separator bar. The statistics include the item's title, type, creation date, size, and storage location. A user can change an item's title in the Item Info slip.

ink text Words written in **electronic ink.**

input Information transferred into a Newton from some external source, such as the pen or a modem. Compare to **output.**

input area A place in a container view where a user can write or draw.

input line An input area where a user can write one line of text.

insertion point See **caret.**

interface See **Newton user interface** and **user interface.**

invert To highlight by changing white pixels to black and vice versa.

large Close box Behaves the same as a **Close box** but looks slightly larger to match the standard height of a text button.

main view A principal container view that serves as a center of user operations for an application. Compare to **application base view.**

matte border A thick gray border framed with black.

message An instruction to execute a **method** (a programmed function).

method A programmed function. Each method contained in a template processes a particular message for the view that the template defines. When a view receives a message for which it has a method, the Newton executes that method.

Names File The built-in application for storing names, addresses, phone numbers, and other information about people.

Newton user interface
The standard conventions for interacting with Newton devices. The interface ensures users a consistent means of interacting with all Newton devices and the applications designed to run on them.

Notepad The built-in application for taking and organizing notes, which may contain text and drawings.

notification slip A view that appears on the screen to warn the user or to report an error. A notification slip may or may not be accompanied by an alert sound.

output Information transferred from a Newton to some external destination, such as a printer or a modem. Compare to **input.**

palette A small view that provides controls for modifying the contents of other views. The user can move a palette, and it floats on top of other views of the same application, so it can remain open for use in all visible views.

parent view A view that contains one or more other views.

persona The permanent internal description of an individual person that works with a particular Newton PDA, or a particular public image of a single owner. The owner is the obvious example, but there can be many others. Choosing a persona sets up information such as name, title, birthday, phone numbers, and e-mail addresses.

picker A list of choices that appears when the user taps an adjacent text label marked by a solid diamond (◆). A picker may also appear when a user taps a **button.** The user chooses one of the listed items by tapping it.

picture button A control that the user taps to designate or confirm an action implied by the icon displayed on the button.

picture radio button

A standard Newton control that displays its state, either on or off, and is part of a group in which the user can turn on only one button at a time. A picture on the on-off indicator identifies the kind of setting.

pixel Short for *picture element;* the smallest dot the Newton can draw on the screen. On a Newton MessagePad, there are 80 pixels to an inch. Each pixel can be either black or white, so it can be represented by a bit; thus, the display is said to be a **bitmap.**

point A unit of measurement for type. 1 point equals approximately $^{1}/_{72}$ inch.

proto template A predefined template that defines the appearance and behavior of a standard interface element, such as a Close box or a status slip. A proto template is called a "proto" for short.

radio button A standard Newton control that displays its state, either on or off, and is part of a group in which the user can turn on only one button at a time. A text label next to the on-off indicator identifies the kind of setting.

routing The process of sending or receiving data through the built-in In/Out Box application using a communications transport such as printing, faxing, beaming, or e-mailing.

routing slip A view in which a user specifies the sender, recipient, format, and other information needed to send data by the method the user picked from the Action picker.

scroll To cause currently displayed data to move off the screen and be replaced by data that was not visible.

scrolling list A boxed list of text items. A user sees more items by tapping the universal scroll arrows or optional local scroll arrows, and selects one or more items by tapping them.

select To designate an object by tapping, double-tapping, or dragging across it. The next action that happens to an object happens to the selected object.

selection The object or group of objects most recently designated to be affected by the next action.

separator bar A heavy black line that heads each item in a view that can display multiple items at once. A separator bar carries the title of the item below it and also carries controls that affect only the one item.

shape A picture composed of geometric shapes such as straight lines and curves, circles and ovals, and rectangles and other polygons.

sibling views Two or more views contained by one other view (their **parent view**).

slider A control with a marker that indicates an amount, degree, or value in relation to a range of possible values. The user can adjust the setting by dragging the marker on a slider. Compare to **gauge.**

slip A matte-framed container view that an application displays to get detailed user input, or to present alternatives among which a user can choose to determine the outcome of a task just begun.

status box A black-framed container view that displays a static message saying the Newton is busy completing a lengthy process.

status slip A view that an application displays when it begins an operation that takes more than a few seconds to complete. A status slip contains a message describing the application's busy status.

system proto See **proto template.**

tap To touch briefly with the pen.

tap-and-a-half To tap and then at the same spot quickly half-tap; the pen goes down, up, and down (but not up again).

template A read-only data structure that precisely specifies a view, encapsulating all the view's attributes and behaviors.

text button A control, bordered by a rounded rectangle, that the user taps to designate, confirm, or cancel an action described by a text label inside the border.

transport A means of conveying data between the built-in In/Out Box application and a communications connection (serial connection, modem, infrared beam, AppleTalk network, and so on).

type ahead The process by which the Newton system stores keystrokes (typed faster than the system can process) for later processing.

view A visual object on the screen, including but not limited to a **container view.** For example, text buttons, pickers, and input areas are also views. Each view is internally represented by a **template.**

user interface The rules and conventions by which a device communicates and interacts with the person operating it.

word wrap The automatic continuation of text from the end of one line to the beginning of the next without breaking in the middle of a word.

Index

THE APPLE PUBLISHING SYSTEM

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh computers and FrameMaker software. Proof pages were created on an Apple LaserWriter Pro 630 printer. Final page negatives were output directly from the text and graphics files. Line art was created using Adobe™ Illustrator. PostScript™, the page-description language for the LaserWriter, was developed by Adobe Systems Incorporated.

Text type is Palatino® and display type is Helvetica®. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Apple Courier.

WRITER
Lon Poole

PROJECT LEADER
Christopher Bey

ILLUSTRATOR
Peggy Kunz

EDITOR
David Schneider

PRODUCTION EDITOR
Rex Wolf

PROJECT MANAGER
Gerry Kane

Special thanks to Marge Boots, Bob Ebert, and Garth Lewis.