

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

This supplement presents user interface guidelines for the enhanced keyboard features in Newton OS 2.1. These guidelines describe:

- **The key view:** what it is and how users spot it
- **The keyboard:** uses of the various keys, including new uses of the return and arrow keys, and uses of the new enter, command, and dedicated function keys
- **Selecting:** methods for text and optional methods for graphics
- **Default action:** how users recognize and operate a view's default button, default Close box, or other default action
- **Navigating through container views:** rotating through input fields, scrolling a view, and dismissing a view
- **Operating pickers:** opening, closing, browsing, and choosing a listed item
- **Keyboard equivalents:** standard keyboard commands defined by and reserved by the system for specific buttons and picker items; keyboard

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

commands defined by the built-in applications; and two methods for reminding users about available keyboard equivalents

The guidelines in this document apply to Newton OS 2.1 devices with built-in keyboards, such as Shay, and to Newton OS 2.1 devices with an attached Newton keyboard. These guidelines do not apply to typing passed through a personal computer to a Newton OS 2.1 device.

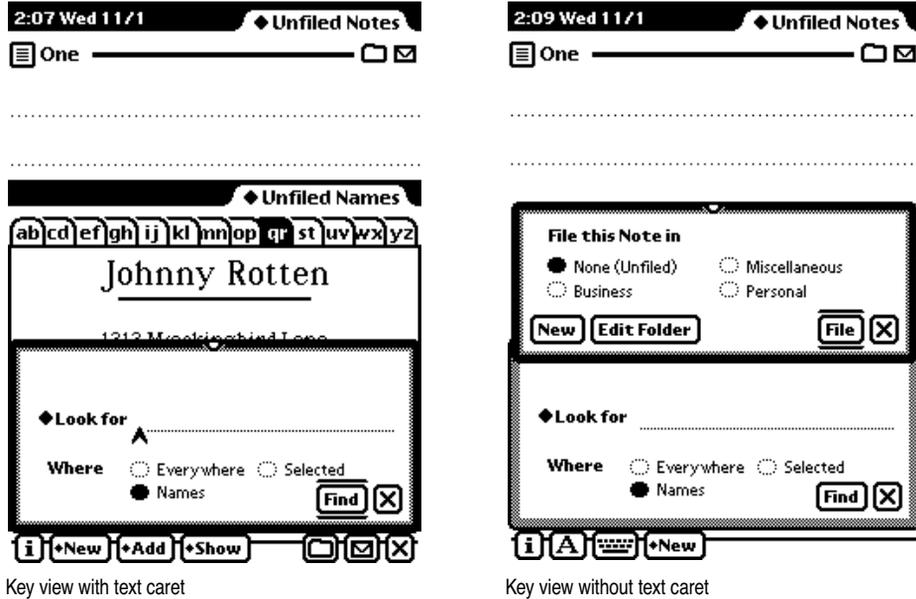
Note

This is a preliminary draft of *User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements*. What you read here is subject to change and may not consistently reflect the latest technical information available, although we have made every effort to supply you with information that is as accurate as possible. ♦

The Key View

Only one container view receives keyboard input, including keyboard commands. This view is called the **key view**. Users can spot the key view by looking for the container view with a heavy black border. In addition, if there is a text-insertion caret or a range of selected text, the caret or text selection marks the key view. Applications should always set the caret or preselect some text when they display a view that has a text input field. That's not to say there is always a caret or a text selection. The key view cannot contain the caret or a text selection if that view has no text input area. In that case, there is no caret or text selection in any view. Even if the key view is not marked with the caret or a text selection, the key view still receives keyboard input. Figure 9-1 compares three key views, one with the caret, one with a text selection, and one with neither mark; in each case the key view has a heavy black border.

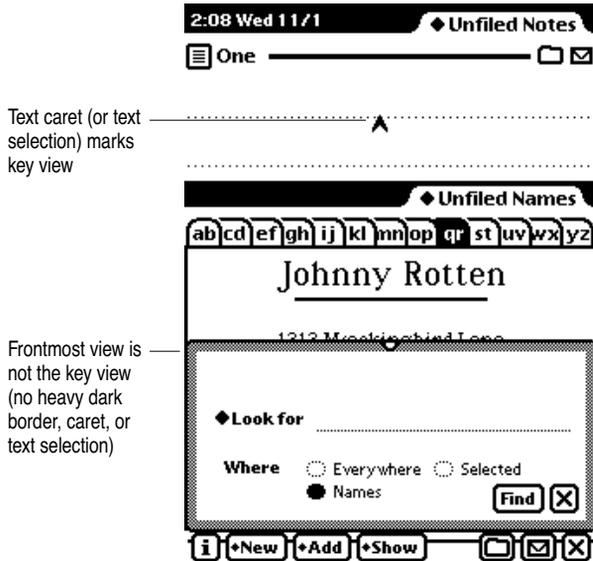
Figure 9-1 A heavy black border marks the key view



Key view with text caret

Key view without text caret

The key view is often the frontmost view, but it does not have to be. A user can place the caret (or make a text selection) in a text input area of any visible view. Figure 9-2 shows an example of the caret in a view other than the frontmost view.

Figure 9-2 The key view can be the frontmost view or not

The Keyboard

In the Newton OS 2.1 interface, users should be able to operate all controls and input all data solely with a pen. A user may attach a keyboard to facilitate entering text, and may use keyboard commands to operate some controls. Keyboard commands are always alternatives to operating controls by tapping with a pen; they should never be the only method of giving a command.

This section discusses what should happen when a user presses the various keys on the keyboard of a Newton OS 2.1 device. In addition, this section describes differences between pressing a key on a hardware keyboard and tapping the same key on an on-screen keyboard. In this section, the keys are

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

organized in three groups: character keys, modifier keys, and editing keys. Character keys input text. Modifier keys redefine most of the character keys. Editing keys move 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 a user presses one of these keys while entering text, the corresponding character is added to the text. The tab, return, and delete (or 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

The effect of pressing the return key depends on the existence and location of the text-insertion caret or text selection. If the caret is in an input field that allows entering multiple lines of text, pressing return inserts a carriage return at the caret. If a user presses return with some text selected in a multiple-line text input field, the selected text is deleted and replaced by a carriage return. The carriage return ends the current paragraph and places the caret at the beginning of the next line.

If the caret is in a one-line text input field, pressing return initiates the key view's default action (described in "Default Action" on page 9-14). This behavior is new with Newton OS 2.1. With earlier versions of the system, pressing return while typing in a one-line text input field moves the caret to the next field in the same container view. The tab key accomplishes that action in Newton OS 2.1 as well as earlier system versions.

If there is no caret, pressing return on a Newton OS 2.1 device's keyboard initiates the default action of the key view (see "Default Action" on page 9-14). With versions of the system prior to Newton OS 2.1, pressing return has no effect when there is no caret.

Enter

The enter key has the same effect as the return key.

Tab

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

Delete (or Del)

The delete (or del) key deletes currently selected text. If no text is selected, tapping the delete key removes the character preceding the text-insertion caret. In a graphics application, the delete key can remove selected graphics objects. The system handles text deletion, but your application has to handle keyboard deletion of graphics itself.

Modifier Keys

Modifier keys are those that alter the way other keystrokes are interpreted. (These keys do not affect tapping with the pen.) They are the shift, caps lock (or caps), option, command, and control keys. It is important that you use these keys consistently from application to application, as outlined in these guidelines.

Shift

The shift key acts like its namesake on a typewriter. When held down at the same time a character key is pressed, the shift key makes alphabetic keys produce capital letters, number keys produce punctuation or symbols, and so on. With Newton OS 2.1, the shift key can also be used in conjunction with the pen to constrain movement in graphics applications. For example,

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

holding down the shift key while drawing a rectangle could limit the pen to drawing squares.

An on-screen keyboard's shift key, unlike the shift key on a hardware keyboard, is not pressed concurrently with pressing a character key. An on-screen keyboard's shift key locks on when tapped and releases automatically when a character key is tapped.

Caps Lock (or Caps)

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

An on-screen keyboard's option key, unlike the option key on a hardware keyboard, is not pressed concurrently with pressing a character key. An on-screen keyboard's 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.

With Newton OS 2.1, the option key can also be used in conjunction with the pen to modify the effect of a tap or drag. For example, if a user selects some text and holds down the option key while dragging the selection, the system makes a copy of the selected text and moves it to wherever the user lifts the pen.

Command

The command key is labeled with a cloverleaf symbol (⌘), an Apple symbol (⌘), or both. Pressing a character key while holding down the command key usually tells the application to interpret the key as a command, not as a character. These combinations are called keyboard equivalents, as described in “Keyboard Equivalents” on page 9-20.

Ctrl (or control)

The ctrl (or control) key is used in terminal-emulation programs to produce a standard non-printing ASCII character. Newton OS 2.1 does not insert those characters into a text input area. For example, typing ctrl-C does not insert anything in the Notepad.

Arrow Keys

The left arrow and right arrow keys move the text-insertion caret left or right one character at a time. The up arrow and down arrow keys move the caret up or down one line of text at a time, aligning the caret at the nearest character boundary and keeping the caret as close as possible to the original horizontal position.

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. The caret appears at the left edge of a text selection when a user presses the left arrow or up arrow key. The caret appears at the right edge of a text selection when a user presses the right arrow or down arrow key. These actions don't move the selected text.

In some cases it's appropriate to use modifier keys such as option and command to move the input focus or extend a text selection. Keep in mind that these key combinations are only shortcuts for pen gestures. Although your application can employ keyboard equivalents to pen-based actions, it should never implement a keyboard method without an equivalent pen-based method.

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

The option key and the command key should be used to intensify the effect of the arrow keys. The command key intensifies the effect more than the option key. For example, in word-processing applications, the unmodified arrow keys move one character or one line at a time; with the option key they move the caret one word or one paragraph at a time; and with the command key they move the caret to the start or end of the current line or body of text. Table 9-1 details how the option key and command key intensify the effect of arrow keys in a word-processing application.

Table 9-1 Modifier keys change the movement of the caret with the arrow keys

	Arrow key alone	With option key	With command key or shift key
Left arrow	Left one character	Left one word	To beginning of line
Right arrow	Right one character	Right one word	To end of line
Up arrow	Up one line	To start of paragraph	To start of first paragraph
Down arrow	Down one line	To end of paragraph	To end of last paragraph

Modifier keys change the movement of the caret with the arrow keys. Though the use of multiple modifier-key combinations (such as command–option–left arrow) is discouraged, it’s all right to use the shift key with command or option to make a text selection. (See “Selecting Text” on page 9-11 for more information.)

Function Keys

Some keyboards include dedicated function keys. Table 9-2 identifies the equivalent button, picker item, or action for each dedicated function key.

Table 9-2 Dedicated function keys

Key	Equivalent action
Close	Tap the Close box
Extras	Tap the Extras button
Dock	Tap the Connection icon
Send	Beam the current item
Receive	Receive a beamed item
Scroll Up	Tap the universal up arrow
Overview	Tap the Overview button
Scroll Down	Tap the universal down arrow
Undo	Tap the Undo button
Find	Tap the Find button
Assist	Tap the Assist button
Light	Turn backlighting on and off
Power	Turn the device on and off

Type-Ahead and Auto-Repeat

If a user types more quickly than the system can handle keystrokes, the system queues the extra keystrokes for later processing. This queuing is called **type-ahead**. There's a limit (varying with the device) 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.

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

When a user holds down a key (or presses the pen on an on-screen keyboard) for a certain amount of time, the system acts as if the user were repeatedly pressing that key. This feature, called **auto-repeat**, affects character keys, modifier keys, and arrow keys. Keyboard commands are auto-repeated as well as text. Auto-repeat does not function during type-ahead. It operates only when the system is ready to accept typing.

Selecting

When performing an operation on an object, a user must select the object to distinguish it from other objects. A user can make any kind of selection with the pen, but can make only some kinds of selections with keyboard commands. These are the selections a user can and can't make with keyboard commands:

- Can select all text
- Can move the text-insertion caret left, right, up, and down
- Can extend or shrink a text selection
- Can't change a selection by holding down a key while tapping with the pen
- Can't set the caret's position arbitrarily
- Can't select graphic objects
- Can't make discontinuous selections
- Can't make an inherent selection while performing an operation (for example, scrubbing an object to delete it)

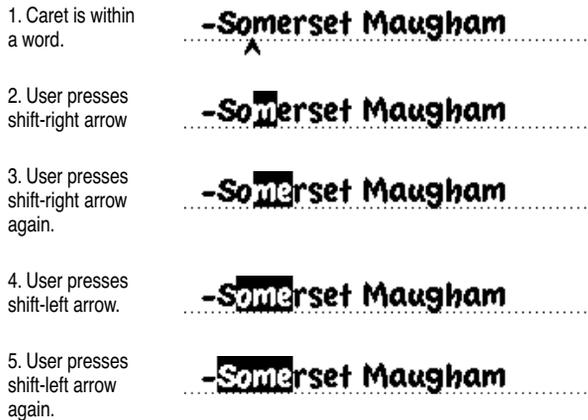
Selecting Text

Users can select text with the keyboard by holding down the shift key while pressing the left- or right-arrow key. The text selection begins at the caret and grows by one character each time a user presses shift-right arrow or shift-left

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

arrow. The selection grows in the direction that the arrow key points. Pressing the other arrow key (while holding down shift) grows the selection from its other end. Note that pressing the other arrow key does not shrink the selection. Figure 9-3 illustrates this series of actions.

Figure 9-3 Selecting with shift and arrow keys



Users can select whole words instead of single characters by pressing option-shift-right arrow or option-shift-left arrow. If the caret is in the middle of a word when a user presses option-shift-right arrow or option-shift-left arrow, the word that contains the caret is selected. Figure 9-4 illustrates selecting word by word with the keyboard.

Figure 9-4 Selecting with option-shift and arrow keys

1. Caret is within a word.

but they only want praise.

2. User presses option-shift-right arrow.

but they only want praise.

3. User presses option-shift-right arrow again.

but they only want praise.

A user can press arrow-key combinations to extend a text selection initially made with a pen. Pressing shift-right arrow or option-shift-right arrow extends the end of the selection to the right. Pressing shift-left arrow or option-shift-left arrow extends the beginning of the selection to the left.

If a user selects some text, either with a pen or with arrow keys, pressing any arrow key alone (without a modifier key) deselects the text and moves the caret. If a user presses left arrow or up arrow, the caret appears at the beginning of what had been selected. If a user presses right arrow or down arrow, the caret appears at the end of what had been selected.

A user can press command-A to select all text in the input area that has the caret. If there is no caret, but the key view has only one input area for text, such as a note in the Notepad, pressing command-A selects all text in the input area. In a view that must be scrolled to see all the text it contains, such as a long note in the Notepad, command-A selects all the text in the view, including the part that can't be seen. As soon as a user scrolls the view, Newton OS 2.1 deselects all the text.

Users can select text with the pen or with keyboard commands, but cannot change what is selected by holding down a key while tapping or dragging with the pen. A user cannot extend or shrink a text selection by pressing shift while tapping.

Selecting Graphics

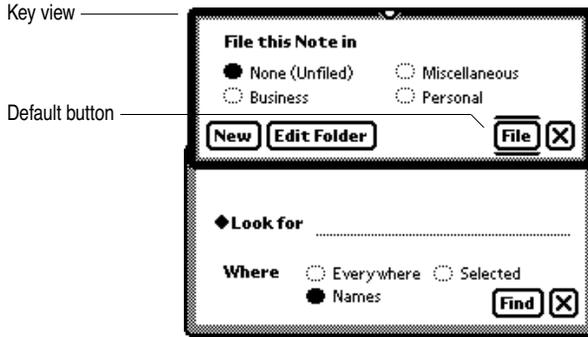
Users can draw shapes and sketches only with a pen. Since users won't be creating graphic objects with a keyboard, there is no compelling reason to allow selecting graphics with a keyboard. Command-A should not select graphic objects that are in the same view as text. Users can select multiple graphic objects with a pen (see "Selecting Text and Shapes" on page 6-22), so they don't need to be able to select multiple graphic objects by tapping each one while pressing the shift key.

A graphics application can allow users to nudge selected graphic objects by tapping arrow keys. This behavior is optional; Newton OS 2.1 doesn't implement it system-wide.

Default Action

In many slips and other container views, it's helpful to assign a standard keyboard command to the completion action that users are most likely to take among the safe alternatives. This action is called the **default action**, and it has two standard keyboard commands: pressing return or enter. Pressing return or enter in a view that is not an overview should correspond to tapping the most frequently used control—close box, text button, or picture button—that initiates a safe action. Pressing return or enter in an overview should switch to a detail view of the item currently highlighted in the overview.

Newton OS 2.1 visually indicates the default button or default close box in a view that has one by drawing heavy black lines above and below the default button or default close box. This visual distinction only appears when a keyboard is actually connected. Figure 9-5 shows an example of a default button and a default close box.

Figure 9-5 Heavy black lines indicate a default button or default close box

Although any view can have a default action, many views don't need one. Don't use a default action if the most likely action is dangerous—for example, if it causes a loss of user data. If a view doesn't have a default button or default close box visibly designated, pressing return should not make it go away; a user must tap a button or the close box. (A user can also press command-W, the standard keyboard command for the Close box.) This guideline protects users from damaging their work by accidentally pressing return or enter. Also, users should be able to predict the effect of pressing return, and that prediction is based on seeing the default button or close box with its distinctive marking. You can consider using a safe default button, such as Cancel (see “Naming Cancel- and Stop-Action Buttons” on page 3-5).

Some slips where users enter text can have a default button or default close box, but some should not. If a slip allows users to enter only single paragraphs of text, then a default action is fine. If a slip allows users to enter multiple paragraphs (for example, to start a new line in a text field by pressing return), then the slip should not have a default button or default close box. A slip that has two behaviors for the return key could confuse users and make the interface seem less predictable.

Keyboard Navigation in Views

With keyboard commands, users can scroll, rotate through input areas, and dismiss slips and other container views.

Rotating Through Input Areas

In a slip or other container view, a user can proceed sequentially through the interface elements that accept keyboard input, such as text input lines, in several ways. In addition to the primary methods—tapping and writing directly in an input field (see Chapter 6, “Data Input”)—a user can press the tab key to rotate through the available input fields. A user can move backward through the available elements by pressing the combination shift-tab.

The system indicates which text input field is currently receiving keyboard input by displaying the text-insertion caret or highlighting text a user has selected.

Scrolling a View

If users can scroll a view by tapping the universal scroll arrows, they should be able to scroll by pressing command-↑ and command-↓. In a view that has local scroll arrows, your application can allow users to perform the same actions by pressing command-shift-↑ and command-shift-↓.

Dismissing a View

To dismiss a key view that has a Close box, a user can always press command-W. If the key view has a Cancel button or a Stop button, a user can dismiss the view by pressing command- . (period). A user can also dismiss a view by pressing return or enter to initiate the view’s default action, which may or may not be the Close box (see “Default Action” on page 9-14).

Picker Operation

In addition to the usual methods for working with pickers, users can operate many pickers from a keyboard. There are keyboard commands for opening a picker and selecting a listed item. There are also keyboard commands for scrolling a long picker and for closing a picker without making a choice. All the keyboard commands work with all list pickers. The keyboard commands for opening and closing work with date pickers, time pickers, number pickers (flippers), or data overview pickers that are associated with text input fields; users can browse through and choose from those types of pickers only with a pen. (For descriptions of the various types of pickers, see Chapter 4, “Pickers.”)

Opening and Closing Pickers

Users can use a keyboard to pop up pickers from text input fields whose label begins a diamond and from some buttons.

Opening Pickers from Buttons

To pop up a list picker from a button, a user presses the command key along with a particular character key that the system or the active application has defined. This key combination is called a **keyboard equivalent**. Newton OS 2.1 defines standard keyboard equivalents for some common buttons, and each application can define other keyboard equivalents for its buttons. The keyboard equivalents defined by Newton OS 2.1 and the built-in applications are listed in “Keyboard Equivalents” on page 9-20.

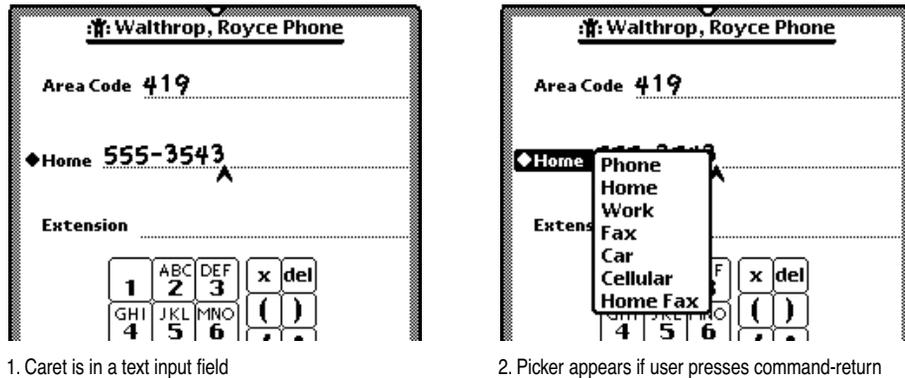
Opening Pickers from Field Labels

To pop up a field label’s picker, a user presses command-return while the text-insertion caret is in the field or while text in the field is selected. Your application probably does not have to define the command-return

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

combination for its label pickers, because Newton OS 2.1 defines the command-return combination for all label pickers created with system proto templates. Figure 9-6 shows how a keyboard command pops up a field's label picker.

Figure 9-6 Opening the current field's label picker



1. Caret is in a text input field

2. Picker appears if user presses command-return

Closing Pickers

Users can close any picker without choosing an item by pressing command-W or command- . (period).

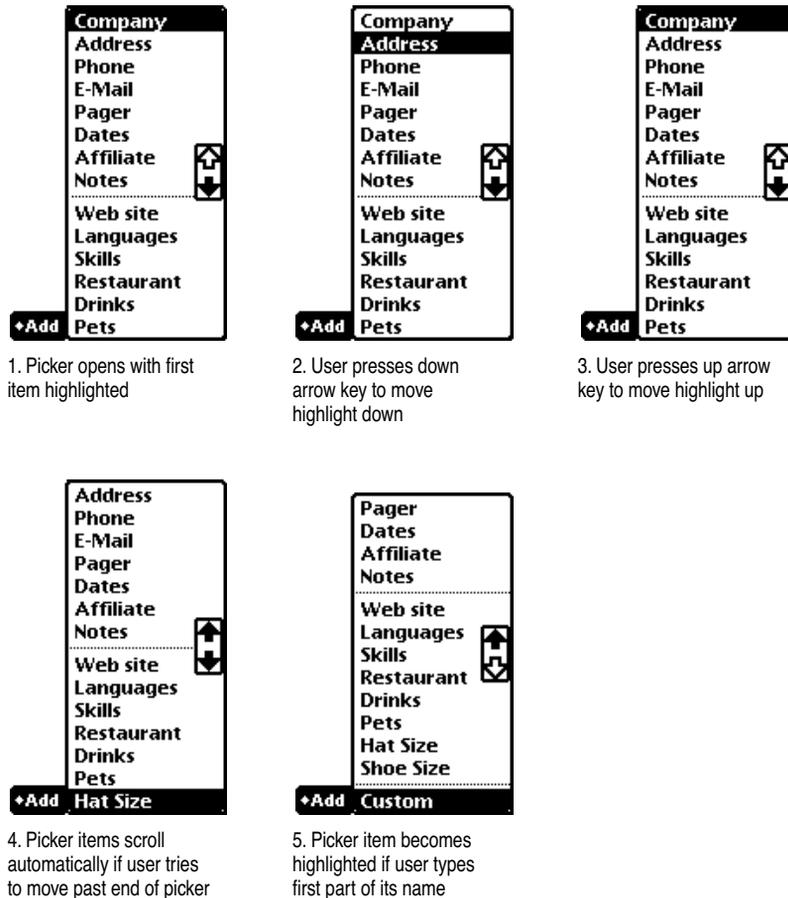
Browsing and Choosing in List Pickers

After opening a list picker, a user can use a keyboard command to choose a listed item or to browse the list of items and dismiss the picker without choosing an item. A list picker opens with one item highlighted. A user can highlight the picker item above or below the currently highlighted item by pressing the up- or down-arrow keys. If necessary, the picker items scroll up or down automatically as a user presses an arrow key to move the highlight at the end of the picker. The item initially highlighted in a picker that pops up from a button is the item at the top of the picker. For a picker that pops up

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

from a field label, the item initially highlighted corresponds to the current value of the field. Figure 9-7 shows how keyboard commands highlight and scroll picker items.

Figure 9-7 Highlighting, scrolling, browsing, and choosing in list pickers



User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

A user can also select a picker item by typing one or more characters that match or alphabetically come before the desired item. If necessary, the picker items scroll automatically to reveal the type-selected item.

Pressing return chooses the highlighted item, the same as tapping that item. To close a picker without picking the highlighted item, a user can press command-period or command-W.

Keyboard Equivalents

The system and applications can define keyboard commands that are the equivalent of tapping buttons or choosing from pickers. In most cases, a keyboard equivalent consists of pressing the command key in combination with a letter, number, punctuation, symbol, or arrow key.

Newton OS 2.1 defines a number of system-wide keyboard equivalents, and this section identifies them. In addition, each application can define its own keyboard equivalents. This section explains how to decide on keyboard equivalents for a specific application, and describes the built-in applications' keyboard equivalents as examples of application-specific keyboard equivalents. In addition, this section discusses how Newton OS 2.1 and applications can help users recall the available keyboard equivalents.

System-Wide Keyboard Equivalents

Table 9-3 lists the system-wide keyboard equivalents defined by Newton OS 2.1. The key combinations in Table 9-3 are reserved across all applications. Your application should use them only for the equivalent buttons, icons, picker items, and actions documented in the table.

Table 9-4 lists some additional reserved keyboard equivalents. The key combinations in Table 9-4 are reserved to preserve continuity between Newton OS 2.1 devices and personal computers that use the Mac OS. (The Mac OS uses those key combinations for switching languages, keyboard

layouts, and input methods.) Your Newton OS 2.1 application should not use the keyboard equivalents documented in Table 9-4.

Table 9-3 System-wide reserved keyboard equivalents

Keys	Equivalent action
Cmd-A	Select all text in the current input area ¹
Cmd-C	Copy selected text to clipboard
Cmd-F	Tap the Find button
Cmd-N	Tap the New button
Cmd-O	Tap the Overview button
Cmd-P	Choose Print from the Action picker
Cmd-R	Tap the Routing (or Action) button
Cmd-S	Tap the Show button
Cmd-V	Paste clipboard text at caret, and clear clipboard
Cmd-option-V	Paste clipboard text at caret without clearing clipboard
Cmd-W	Tap the Close box
Cmd-X	Cut selected text to clipboard
Cmd-Z	Tap the Undo button
Cmd-return	Tap the current input area's ¹ diamond label, or tap the checkmark of the current overview or list item
Cmd- ` (tilde)	Tap the Assist button
Cmd- . (period)	Tap the Cancel button, Stop button, or Close box
Cmd-shift-F	Tap the Filing button
Cmd-shift-S	Tap the Styles icon
Cmd-shift-T	Tap the Folder tab
Cmd- ?	Tap the How do I? button or choose Help from the Info picker

Table 9-3 System-wide reserved keyboard equivalents

Keys	Equivalent action
Cmd- /	Tap the How do I? button or choose Help from the Info picker
Cmd- ↑	Tap the universal up arrow ²
Cmd- ↓	Tap the universal down arrow ²

NOTE ¹The current input area is the one that has the text-insertion caret or text selection

NOTE ²In word processing applications, Cmd- ↑ or Cmd- ↓ scrolls to the beginning or end of the key view

Table 9-4 Additional reserved keyboard equivalents

Keys	Equivalent action
Cmd-space bar	Reserved for worldwide systems
Cmd- <i>modifier</i> key-space bar	Reserved for worldwide systems
Cmd- →	Reserved for worldwide systems
Cmd- ←	Reserved for worldwide systems

Common Keyboard Equivalents

Some applications use other common keyboard equivalents that are not reserved, as shown in Table 9-5. If your product has a function listed in this table, it should use the keyboard equivalent listed in the table. If your

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

product doesn't support one of these functions, then use that function's keyboard equivalent as you wish.

Table 9-5 Common keyboard equivalents that are not reserved

Keys	Equivalent action
Cmd-T	Set Plain text style
Cmd-B	Set Bold text style
Cmd-I	Set Italic text style
Cmd-U	Set Underline text style
Cmd-shift-↓	Tap a local down arrow
Cmd-shift-↑	Tap a local up arrow
Cmd- =	Initiate a spelling check
Cmd-E	Tap an Item Info button

Application-Specific Keyboard Equivalents

Your application should respond to the standard system-wide keyboard equivalents as much as possible. In addition, your application can define keyboard equivalents to operate its own buttons and picker items. As you decide which buttons and items to provide equivalents for, follow these guidelines:

- Define keyboard equivalents only for frequently used buttons and picker items. Don't clutter the interface with keyboard equivalents. In general, picker items should not have their own keyboard equivalents. A user can always select a picker item by pressing the arrow keys or by type-selecting.
- Define keyboard equivalents that minimize switching from keyboard to pen and vice versa. Switching input devices throws off a user's tempo and makes your application seem harder to use.
- If possible, use a key combination that matches the first letter of the name of the equivalent button or picker item. For example, command-N pops

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

up the New picker, if there is one. If another keyboard equivalent already uses that key combination, try to use some other mnemonic combination. For example, several built-in applications use command-+ for the Add button, since command-A is already defined by Newton OS 2.1 as Select All.

- Do not use command-Q to “quit” your application. If a user wants to make your application go away, the user should close all its views—by tapping their close boxes or pressing command-W repeatedly.
- Wherever possible, make keyboard equivalents work the same with the shift key pressed or not pressed—don’t simply leave the uppercase variation undefined. If you define different keyboard equivalents with and without the shift key pressed, try to make it possible for users to predict when to use the shift key.
- Avoid keyboard equivalents that require pressing several modifier keys. Generally, combinations of the command key and a character key are the easiest for users to remember and type; combinations that require command plus shift or option are less mnemonic; and combinations that require command, shift, and option are most forgettable.
- Stay consistent with keyboard equivalents used in Mac OS and Windows applications where possible.
- Abide by the reserved and common keyboard equivalents listed in Table 9-2, Table 9-3, and Table 9-4.

The standard keyboard equivalents take a minimalist approach that allows extensibility and flexibility. For example, there are system-wide keyboard equivalents for the New, Show, and Add buttons, but not for items in the New, Show, and Add pickers. (The alternative approach—providing command key equivalents for each type of stationery and viewing option—asks users to remember too many keyboard equivalents.) As a further example, notice that Newton OS 2.1 only defines a keyboard equivalent for one picker item, Print. There are no keyboard equivalents for other routing actions, such as Fax and Mail, because those actions can’t be completed without a pen (to specify an address and set various options). It doesn’t make a lot of sense to start users down the road of keyboard equivalents only to force them to pick up the pen. Print is another story. Users will not need a pen to complete the Print action unless they need to change printers. Also,

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

command-P has become a de facto standard keyboard equivalent for the Print command, but no such standards exist yet for Fax and Mail commands.

Don't let over-reliance on keyboard commands ruin the integrity of Newton OS 2.1's pen-based user interface. Optimize for the pen, and incorporate a keyboard only where it makes sense.

Built-in Applications' Keyboard Equivalents

In addition to following the abstract guidelines presented in the preceding section, your application should be consistent with the specific keyboard commands used in the built-in applications where applicable. If your application has the same button or picker item as a built-in application, then your application should have the same keyboard equivalent. For example, if your application has an Add button, it should recognize cmd- + (and cmd- =) as keyboard equivalents. If your application does not have an Add button, it could use cmd- + as the keyboard equivalent of a different button or picker item. This section details the keyboard commands in the built-in Notepad, Names File, Date Book, Call Log, Filing Slip, Bookplayer, and Calculator.

Notepad

The Notepad application defines the keyboard equivalents listed in Table 9-6 for its Checklist and Outline types of notes.

Table 9-6 Keyboard equivalents for Notepad Checklist and Outline notes

Keys	Equivalent action
Cmd- =	Create a new topic at the current level
Cmd-]	Create a new topic one level to the right
Cmd- [Create a new topic one level to the left
Cmd-return	Check or uncheck the current topic's checkbox (checklist notes only)

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

Note that there are no keyboard equivalents for demoting, promoting, expanding, or collapsing topics. Users must manipulate topics with the pen.

Names File

The Names File application is difficult to adapt for a keyboard because it has so many slips for entering and editing data. It has two application-specific keyboard equivalents, listed in Table 9-7, and relies heavily on several system-wide keyboard equivalents.

Table 9-7 Keyboard equivalents for Names File

Keys	Equivalent action
Cmd- +	Tap the Add button
Cmd- =	Tap the Add button

Command- + was chosen for the Add button because the natural choice, command-A, is reserved system-wide for Select All. Note that the keyboard equivalent works whether or not a user holds down the shift key. The unshifted form on a U.S. keyboard, cmd- =, is commonly used to initiate a spelling check, but the Names File redefines the key combination since it does not have the spelling-check function.

After popping up the Add picker, a user can use the up- and down-arrow keys to select an input slip listed in the Add picker (or can select a picker item by typing its name), and then can open the input slip by pressing the return key.

With an input slip open and the caret or text insertion in a labeled input field, the system-wide keyboard command-space bar lets a user pop up the labeled field's picker. The user can select a picker item with the up- and down-arrow keys or by typing the item's name, and can choose the selected item by pressing return.

Other system-wide keyboard commands let a user go to the next or previous input field by pressing the tab or shift-tab, and put the slip away by pressing

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

the return key. The tab and shift-tab keyboard commands are circular; pressing tab in the last field goes to the first field, and pressing shift-tab in the first field goes to the last field.

Date Book

The Date Book application presents unique challenges for incorporating a keyboard because it uses date pickers, time pickers, and number pickers (flippers) extensively. There simply isn't an obvious way to operate those controls with the keyboard. The system-wide keyboard equivalents let users operate the New and Show pickers by keyboard. Table 9-8 lists the Date Book's specific keyboard equivalents.

Table 9-8 Keyboard equivalents for Date Book

Keys	Equivalent action
Cmd- +	Tap the Add button
Cmd- =	Tap the Add button

Although the date, time, and number pickers used in the Date Book don't lend themselves to keyboard commands, the Date Book application has been revised to accommodate keyboard input. For example, a meeting or event slip now opens with the text-insertion caret in the Title field to facilitate typing.

Call Log

The Call Log application defines the keyboard equivalents listed in Table 9-9.

Table 9-9 Keyboard equivalents for Call Log

Keys	Equivalent action
Cmd- +	Tap the Add button
Cmd- =	Tap the Add button
Cmd-D	Tap the Place Call button (Dial button)
Cmd-H	Tap the Hang Up button
Cmd-N	Tap the New button

Filing Slip

When the Filing slip is open, the system defines the keyboard equivalents listed in Table 9-10.

Table 9-10 Keyboard equivalents for Filing Slip

Keys	Equivalent action
Cmd-E	Tap the Edit Folder button
Cmd-N	Tap the New button

BookPlayer

The BookPlayer application defines the keyboard equivalents listed in Table 9-11

Table 9-11 Keyboard equivalents for BookPlayer

Keys	Equivalent action
Cmd-G	Tap the Page Number button
Cmd-B	Tap the Bookmark picture button
Cmd-M	Tap the Markup picture button

Calculator

In the Calculator, users can type numbers and arithmetic operators instead of tapping the application's integral on-screen keypad. Table 9-12 lists the Calculator's keyboard equivalents, which do not require holding down the command key concurrently.

Table 9-12 Keyboard equivalents for Calculator

Keys	Equivalent action
<i>Any number</i>	Tap the corresponding number button
. (period)	Tap the . (decimal point) button
=	Tap the = button
Return	Tap the = button
Enter	Tap the = button
C	Tap the C (clear) button
E	Tap the CE (clear entry) button
Delete	Tap the CE (clear entry) button

Table 9-12 Keyboard equivalents for Calculator

Keys	Equivalent action
+	Tap the + button
-	Tap the – button
/	Tap the ÷ button
X	Tap the × button
*	Tap the × button
%	Tap the % button
√ (option-V)	Tap the √ button

Viewing Available Keyboard Equivalents

Newton OS 2.1 includes two on-screen mechanisms for reminding users about available keyboard equivalents. The primary mechanism is a directory of keyboard equivalents, which the system displays in a pop-up view when a user holds down the command key for about one second. The directory shows keyboard equivalents that are valid in the current context. The context is determined by what container view is receiving keyboard input and where the caret is positioned, if it's present. Figure 9-8 shows an example directory of keyboard equivalents.

Figure 9-8 A directory lists available keyboard equivalents

Creating			
New	⌘N	Title	⌘T
Editing			
Copy	⌘C	Paste	⌘V
Cut	⌘X	Paste a...	⌘⇧V
		Select All	⌘A
		Styles	⌘⇧S
Filing			
File	⌘⇧F	Folder	⌘⇧T
General			
Assist	⌘`	Find	⌘F
Close	⌘.	Help	⌘/
		Undo	⌘Z
Routing			
Print	⌘P	Routing	⌘R
Viewing			
Close	⌘W	Overview	⌘O
		Show	⌘S

Your application does not display the directory, but it can register its keyboard equivalents for inclusion in the directory. Not every keyboard equivalent has to appear in the directory. A keyboard equivalent does not appear in the directory if it lacks a name like Copy or Find. For example, the keyboard equivalent for the universal up arrow, command-up arrow, does not have a name and is not listed in the pop-up directory.

The system builds the directory dynamically, and groups the keyboard equivalents by category. The directory displays only the keyboard equivalents available in the current context. Newton OS 2.1's system-wide keyboard equivalents fall into the following categories:

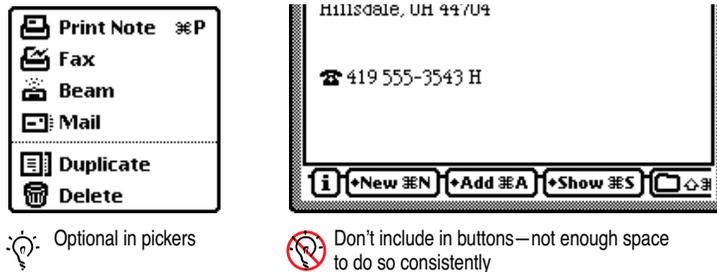
- Creating
- Editing
- Filing
- General
- Routing
- Viewing

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

Your application can add its commands to the standard categories or create its own categories. If a keyboard command has no category but does have a name, it is automatically placed in an “Other” category.

In addition, your application can show keyboard equivalents in pickers. An application should not include keyboard equivalents in button names, because there is usually not enough screen space to do that consistently throughout an application. There certainly is not enough space to show keyboard equivalents for all buttons in every application. Figure 9-9 shows where an application can display keyboard equivalents.

Figure 9-9 Keyboard equivalents are OK in pickers but not in buttons



The Keyboard Equivalents pop-up is displayed as long as a user holds down the command key, and goes away when the user releases the key. Neither the Keyboard Equivalents pop-up nor the keyboard equivalents in pickers appear unless a keyboard is attached to the Newton OS 2.1 device.

Glossary Addendum

auto-repeat

The repeated automatic generation of characters that happens when a user holds down a key (or presses the pen on an on-screen keyboard).

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements

default action The completion action that users are most likely to take among the safe alternatives in the key view.

key view The container view that receives keyboard input and contains the text-insertion caret or text selection (if either exists).

keyboard equivalent A keyboard command that is the equivalent of tapping a button or choosing from a picker.

User Interface Guidelines for Newton OS 2.1 Keyboard Enhancements