# NCKey

By: Ramana Rao (Rao.pa@Xerox.COM)

Stored: {qv}<notecards>1.3k>library>NCKey, .dcom, .ted
Last updated: July 13, 1987.
Requires: TEditKey Lispusers Package

## INTRODUCTION

This NoteCards library package provides an extensible keyboard interface to useful Notecard Operations (useful means that I needed them at the time). In particular, NCKey provides keystroke commands for creating a link to a new text card using the current selection as either the title or the text of the new card (I was taking a large document and bashing it into a notecards hierarchy, so you can see why these were useful), entitling a card using the current selection, deleting all the "DeletedLinkIcon", and coercing the type and displaymode of all the links in the current card. However, the particular functions provided are not so important as the ability to easily bind your own functions to keys. Possibilities for functions as demonstrated by my bindings (the default bindings) include using available functions in NCHACKS or writing your own (or begging, asking, or demanding some one else to write) functions using the NC Programmer's Interface.

## USING THE KEYBOARD INTERFACE

NCKey requires TEditKey but doesn't expect that it be previously loaded. NCKey actions are invoked by typing Meta-W followed by a single character. Currently, the single character can not be Meta-ed (but I realize that this is a pain since you want to be able to keep holding down the Meta key while striking the second character). The mapping from dispatch character to an action routine is made using *NCKEY-BINDINGS*. The default mapping is the following:

| | |
|---|---|
| Inserts a Link Using Selection as Title | (J j) |
| Inserts a Link Using Selection as Text | (K k) |
| Removes Deleted Link Icons | (R r) |
| Sets Title Using Selection | (T t) |
| Coerces Links using Global Parameters | (L l) |
| Brings Up Help Menu | (?) |

These items and their operation are described in greater detail below.

## CUSTOMIZING OR EXTENDING THE KEYBOARD INTERFACE

The provided mechanism isn't particularly elaborate or elegant, but it's basically good enough for most purposes. You can edit the variable *NCKEY-BINDINGS* to remap the bindings to the provided action routines or to bind new action routines. This variable is INITVAR-red by this package, so users can put their own binding in their INIT files. Below is a specification of the syntax of *NCKEY-BINDINGS*:

*NCKEY-BINDINGS* = (<Binding Triple>*)

<Binding Triple> = (<Key Group> <Action-Fn> <Description String>) | NIL

A NIL <Binding Triple> is used to indicate a space in the Help menu.

<Key Group> = <Key> or (<Key>*)

<Action-Fn> = <function> or (<function> <Action-arg>*).

In the first case, the function is applied to a default set of parameters (<function> CARD NOTEFILE TEXTSTREAM SEL). In the second case, the form is eval-led.

<Action-arg> = The keywords CARD, NOTEFILE, TEXTSTREAM, TEXTOBJ, or SEL or an evalable form using these keywords as arguments. For example, either of the following may be be used as an Action-Fn Item:

(RandomFunction CARD NOTEFILE)

(AnotherRandomFunction (HelloFn SEL) TEXTOBJ).

<Description String> is used in the Help menu (Meta-W ?) to indicate what the key is bound to.

(NCKEY-BUILD-HELP)

After editting *NCKEY-BINDINGS*, you must call this function to update the Help menu (Meta-W ?).

**PROVIDED ACTION-FNS AND PARAMETERS**

*NCKEY-DEFAULT-CARDTYPE* = Text

*NCKEY-DEFAULT-TITLE* = "Untitled"

*NCKEY-DEFAULT-LINKTYPE* = Unspecified

*NCKEY-DEFAULT-DISPLAYMODE* = NIL

These four user parameters are used by the Provided Action-fns as default values. There is a tradeoff between just using default value and prompting the user for values. I opted for default values because 1) they're a little easier and 2) a major point of this was to get something keyboardishly quick. All of these user parameters are INITVAR-red by this package, so users can put their own bindings in their INIT files.

NCKEY-CREATE-LINK-WITH-TITLE

NCKEY-CREATE-LINK-WITH-TEXT

These create a new card and insert a link to this card into the current card. The current selection is used as either the title or the text of the new card, and is also deleted from the current card.

NCKEY-MAP-LINKS

This function coerces the type and displaymode of all the links in the current card using *NCKEY-DEFAULT-LINKTYPE* and *NCKEY-DEFAULT-DISPLAYMODE* .

(NCHACKS.RemoveDeletedIconsFromTextCards CARD)

This function is obviously a part of NCHACKS and demonstrates an Action-FN which takes arguments other than the default arguments. It removes all the delete icons from the current card.

NCKEY-ENTITLE

This function sets the title of the current card using the current selection. It leaves the current selection alone.

NCKEY-HELP

This function brings up the help menu of the current key bindings. Remember that (NCKEY-BUILD-HELP) should be called to update the help menu after editting the key bindings.

## KNOWN PROBLEMS

As mentioned above, the dispatch character can't be a META-character.