0.1 PROMPTFORWORD

PROMPTFORWORD is a function that reads in a sequence of characters, generally from the keyboard, without involving READ-like syntax. The intent is to mimic the prompted-read used by the Alto Exec when asking for login names, passwords etc. Thus a user can supply a prompting string, as well as a ‘‘candidate’’ string, which is printed and used if the user types only a word terminator character (or doesn’t type anything before a given time limit). As soon as any characters are typed the ‘‘candidate’’ string is erased and the new input takes its place.

PROMPTFORWORD accepts user type-in until one of the ‘‘word terminator’’ characters is typed. Normally, the word terminator characters are EOL, ESCAPE, LF, SPACE, or TAB. This list can be changed using the TERMINCHAR.LST argument to PROMPTFORWORD, for example if it is desirable to allow the user to input lines including spaces.

PROMPTFORWORD also recognizes the following special characters:

Control-A, BS, or DEL
Any of these characters deletes the last character typed and appropriately erases it from the echo stream if it is a displaystream.

Control-W or Control-Q
Erases all the type-in so far.

Control-R
Reprints the accumulated string.

?
Calls up a ‘‘help’’ facility. The action taken is defined by the GENERATE?LIST.FN argument to PROMPTFORWORD (see below). Normally, this prints a list of possible candidates.

Control-V
‘‘Quotes’’ the next character: after typing Control-V, the next character typed is added to the accumulated string, regardless of any special meaning it has. Allows the user to include editing characters and word terminator characters in the accumulated string.

(PROMPTFORWORD PROMPT.STR CANDIDATE.STR GENERATE?LIST.FN ECHO.CHANNEL DONTECHOTYPEIN.FLG TIMELIMIT.secs TERMINCHARS.LST KEYBD.CHANNEL OLDSTRING) [Function]

PROMPTFORWORD has a multiplicity of features, which are specified through a rather large number of input arguments, but the default settings for them (i.e., when they aren’t given, or are given as NIL) is such to minimize the number needed in the average case, and an attempt has been made to order the more frequently non-defaulted arguments at the first of the argument list. The default input and echo are both to the terminal; the terminal table in effect during input allows most control characters to be INDICATE’d.

PROMPTFORWORD returns NIL if a null string is typed; this would occur when no candidate is given and only a terminator is typed, or when the candidate is erased and a terminator is typed with no other input still un-erased. In all other cases, PROMPTFORWORD returns a string.

PROMPTFORWORD uses a MONITORLOCK (see page X.XX) so that a second call cannot be started before the first one finished; primarily this is to limit confusion between multiple processes that might try to access the keyboard at the same time, or print in the prompt window ‘‘at the same time’’

PROMPTFORWORD

PROMPTFORWORD is controlled through the following arguments:

PROMPT.STR

If non-NIL, this is coerced to a string and used for prompting; an additional space is output after this string.

CANDIDATE.STR

If non-NIL, this is coerced to a string and offered as initial contents of the input buffer.

GENERATE?LIST.FN

If non-NIL, this is either a string to be printed out for help, or a function to be applied to PROMPT.STR and CANDIDATE.STR (after both have been coerced to strings), and which should return a list of potential candidates. The help string or list of potential candidates will then be printed on a separate line, the prompt will be restarted, and any type-in will be re-echoed.

Note: If GENERATE?LIST.FN is a function, its value list will be ‘‘cached’’ so that it will be run at most once per call to PROMPTFORWORD.

ECHO.CHANNEL

Coerced to an output stream; NIL defaults to T, the ‘‘terminal output stream’’, normally (TTYDISPLAYSTREAM). To achieve echoing to the ‘‘current output file’’, use (GETSTREAM NIL ’OUTPUT). If echo is to a display stream, it will have a flashing caret showing where the next input is to be echoed.

DONTECHOTYPEIN.FLG

If T, there is no echoing of the input characters. If the value of DONTECHOTYPEIN.FLG is a single-character atom or string, that character is echoed instead of the actual input. For example, LOGIN prompts for a password with DONTECHOTYPEIN.FLG being ‘‘*’’.

TIMELIMIT.secs

If non-NIL, this is the number of seconds (as an integer) that the caller is is willing to wait with no input from KEYBD.CHANNEL (see below); if timeout is reached, then CANDIDATE.WORD is returned, regardless of any other type-in activity.

TERMINCHAR.LST

This is list of ‘‘word terminators’’; it defaults to (CHARCODE (EOL ESCAPE LF SPACE TAB)).

KEYBD.CHANNEL

If non-NIL, this is coerced to a stream, and the input bytes are taken from that stream. NIL defaults to the keyboard input stream. Note that this is *not* the same as T, which is a *buffered* keyboard input stream, not suitable for use with PROMPTFORWORD.

OLDSTRING

If non-NIL, this must be a string, which will be destructively used to return the answer.

Examples:

```
(PROMPTFORWORD
   "What is your FOO word?" ’Mumble
   (FUNCTION (LAMBDA () ’(Grumble Bletch)))
   PROMPTWINDOW NIL 30)
```

This first prompts the user for input by printing the first argument as a prompt into PROMPTWINDOW; then the proffered default answer, ‘‘Mumble’’, is printed out and the caret starts flashing just after it to indicate that the upcoming input will be echoed there. If the user fails to complete a word within 30 seconds, then the result will be the string "Mumble".

```
(FRESHLINE T)
(LIST
   (PROMPTFORWORD
      (CONCAT "{"  HOST  "} Login:")
      (USERNAME NIL NIL T))
   (PROMPTFORWORD
      " (password)" NIL NIL NIL ’*))
```

This first prompts in whatever window is currently (TTYDISPLAYSTREAM), and then takes in a username; the second call prompts with " (password)" and takes in another word (the password) *without* proffering a candidate, echoing the typed-in characters as ‘‘*’’.

PROMPTFORWORD