

WS Dialog

andrei

April 17, 2017

1 Description

WS-style dialog

```
<message>: <enter text>
  *RETURN* done | *Backspace* or *^H* erase left
  *^U* cancel | *Del* or *^G* erase char
<Line4>
```

1.1 Main Mode

- On init: display first 3 lines
- On ^U: exit dialog (run restore function)
- On ^M/Enter: run accept function, run restore function

Accept function: exits or enters one of the Line4 modes, based on the input.

1.2 Line4 modes

Line4 modes are used if the input cannot be accepted. They display an error message and wait for input

- ^U/: exit Line4 mode, hide Line4, return to prompt
- Other keys can be used to run a custom function
- Or can be set to use text input + ^M/Enter => accept function

1.3 Example: block save dialog

- message= Write to file
- accept function: try to save a string as a file, if file exists, then enter *Line4-exists* mode, if write error, enter *Line4-write-error* mode.

1.3.1 Line4-exists mode

- message: #That file already exists.# Overwrite (Y/N)? ¹
- y/Y: same as the main accept function
- n/N: same as ^U (delete Line4, return to prompt)

1.3.2 Line4-write-error mode

- message: #Error writing file <fn>.# Press Enter to continue.
- ^M/Enter/^U: delete Line4, return to prompt

1.4 Interface

Every dialog is a mode and every Line4 of a dialog is a mode. Every dialog has a string identifier, which is used to identify the dialog mode: `wsdialog_<dialog-id>` (further: <DID>). All the <DID> must be stored in the `wsdialog_modes` array.

Every Line4 mode of a dialog is identified as <DID>_<id> (further: <l4id>). All the <l4id> must be stored in a <DID>_modes array.

<DID> and <l4id> must be non-null strings.

1.4.1 Message

<DID>_msg variable

1.4.2 Line4 modes

- message: <l4id>_msg variables
- enter string and ^M/Enter: accept function: <l4id>_accept variable contains accept function name. If defined, string mode, otherwise read key mode.

¹#<string># means that the string format is **standout**.

- single key press corresponding different functions:

stored in an associative array, `<l4id>_fn` where the key is the key sequence or combination to be used with `bindkey` for function identification, and the value is the function name.

1.5 Configuration

1.5.1 Create a dialog

- `<DID>` is the name of the dialog
- add dialog to `wsdialog_modes` array
- `<DID>_msg` is the message
- `<DID>_accept` is the accept function
- `<DID>_restore` is the restore function, called after dialog has finished, on success or on cancel

1.5.2 Define Line4 modes

For each Line4

- choose a name: `<DID>_<name>` is the Line4 identifier `<l4id>`
- Line4 modes can be of two different types, determined by whether the `<l4id>_accept` function is defined or not:
 - if it is an *accept* type Line4, define an accept function and store it in `<l4id>_accept`
 - if it is a *key* type Line4, define for each key sequence `<key>` a function `<func>` and store them in the `<l4id>_fns` array as key-value pairs