

```

# -----
# Application initialization script.
# author      William J Giddings, 13/04/08
# version     0.0.1
# -----

#!/bin/sh
# the next line restarts using tclsh \
exec tclsh "$0" "$@"

package require Gnocl

# counter for tags
set lastTag 1
set activeText ""
set activeBlock ""

# create blocks toolBar
set blocksToolBar [gnocl::toolBar]
$blocksToolBar add item -text "%#New" -tooltip "Add new block" -onClicked {set lastTag [addBlock]}
$blocksToolBar add item -text "%#Convert" -tooltip "Modfity Current Block"
$blocksToolBar add item -text "%#GoUp" -tooltip "Goto Previous Block"
$blocksToolBar add item -text "%#GoDown" -tooltip "Goto Next Block"
$blocksToolBar add item -text "%#Delete" -tooltip "Delete Block"
$blocksToolBar add space
$blocksToolBar add widget [gnocl::entry -variable activeBlock ]

# create text widgets
set text(sl) [gnocl::text -baseColor #ffffdd -wrapMode word -onButtonPress {
    set activeText %w
}]
set text(tl) [gnocl::text -baseColor #ffffff -wrapMode word -onButtonPress {set activeText %w}]
set txtBox [gnocl::box]

$txtBox add [list $text(sl) $text(tl)] -fill {1 1} -expand 1

set guiBox [gnocl::box -orientation vertical]

$guiBox add $blocksToolBar -fill {1 0} -padding 0
$guiBox add $txtBox -fill {1 1} -expand 1 -padding 0

set box2 [gnocl::box -orientation vertical]
$box2 add [gnocl::radioButton -text "%#GoUp" -indicatorOn no -onValue cats -variable animal -onToggled {puts "cats"}]
$box2 add [gnocl::radioButton -text "%#GoBack" -onValue dogs -variable animal -onToggled {puts "dogs"}]
$box2 add [gnocl::radioButton -text "%#GoForward" -onValue birds -variable animal -onToggled {puts "birds"}]

$guiBox add $box2

set topLevel [gnocl::window -title "Matched Text Blocks" -child $guiBox]

set blocks ""

# w widget name
# i index of tag to add
# a name of active text, default = NULL

```

```
proc addBlockTag {w i {a {}}} {  
    upvar $w txts  
    foreach item [array names txts] {  
        $txts($item) tag create block_${i} -background cyan -onEvent {setBlockVariable a  
    }  
    if {$a != ""} {  
        $a tag apply selectionStart selectionEnd -tags block_${i}  
    }  
    return [incr i]  
}  
  
# a variable used by spin  
# b  
proc setBlockVariable {a b} {  
    upvar $a var  
    # upvar $b b  
    puts "b = $b"  
    foreach {i j} [split $b _ ] {  
        puts "j = $j"  
        set var $j  
    }  
}  
  
proc addNewBlock {w } {  
}
```