## -\*- tcl -\*- # ### ### ### ######### ######### ######### ## # $Id: idx.html,v 1.8 2007/03/20 05:06:35 andreas\_kupries Exp $ # # Index Formatting Engine : docidx --> HTML. # Single-pass # # Copyright (c) 2003-2007 Andreas Kupries # Freely redistributable. # ### ### ### ######### ######### ######### ## Requisites dt\_source \_idx\_common.tcl dt\_source \_html.tcl # ### ### ### ######### ######### ######### ## API implementation rename idx\_postprocess {} rename fmt\_postprocess idx\_postprocess proc fmt\_plain\_text {text} {return {}} proc fmt\_index\_begin {l t} { global la ti set la $l set ti $t return {} } proc fmt\_key {text} { global key lk ch set lk $text set key($lk) {} set ch([F $lk]) . return {} } proc fmt\_manpage {f l} {Ref [dt\_fmap $f] $l} proc fmt\_url {u l} {Ref $u $l} proc fmt\_index\_end {} { LoadKwid set lines {} BeginHeader ; Meta ; EndHeader BeginBody ; Title ; Navbar BeginIndex ; Keys ; EndIndex EndBody return [join $lines \n] } # ### ### ### ######### ######### ######### ## Helper commands proc Ref {r l} { global key lk lappend key($lk) $r $l return {} } proc F {text} { return [string toupper [string index $text 0]] } proc LoadKwid {} { global kwid # Engine parameter - load predefined keyword anchors. set ki [Get kwid] if {![llength $ki]} return array set kwid $ki return } proc BeginHeader {} { global la ti upvar 1 lines lines lappend lines [markup ] lappend lines [ht\_comment [c\_provenance]] lappend lines [ht\_comment "CVS: \$Id\$ $la"] lappend lines [markup ] lappend lines "[markup ]" return } proc Meta {} { # Engine parameter - insert 'meta' set meta [Get meta] if {$meta == {}} return upvar 1 lines lines lappend lines [markup $meta] return } proc EndHeader {} { upvar 1 lines lines lappend lines [markup ] return } proc BeginBody {} { upvar 1 lines lines lappend lines [markup ] # Engine parameter - insert 'meta' set header [Get header] if {$header == {}} return lappend lines [markup $header] return } proc Title {} { global la ti upvar 1 lines lines if {($la != {}) && ($ti != {})} { set title "$la -- $ti" } elseif {$la != {}} { set title $la } elseif {$ti != {}} { set title $ti } lappend lines "[markup

### ] $title [markup

]" return } proc Navbar {} { global ch cnt dot upvar 1 lines lines set nav {} foreach c [lsort -dict [array names ch]] { set ref c[incr cnt] set ch($c) $ref lappend nav [ALink $ref $c] } lappend lines [markup "

"] lappend lines [join $nav $dot] lappend lines [markup

] return } proc BeginIndex {} { upvar 1 lines lines lappend lines [markup ""] return } proc Keys {} { global key upvar 1 lines lines set lc {} foreach k [lsort -dict [array names key]] { set c [F $k] ; if {$lc != $c} { Section $c ; set lc $c } BeginKey $k References $k EndKey } return } proc Section {c} { global ch upvar 1 lines lines lappend lines [markup {] return } proc BeginKey {k} { upvar 1 lines lines lappend lines [markup ""] lappend lines [BeginColLeft][SetAnchor $k][markup ] lappend lines [BeginColRight] return } proc EndKey {} { upvar 1 lines lines lappend lines [markup ] return } proc References {k} { global key dot upvar 1 lines lines set refs {} foreach {ref label} $key($k) { lappend refs [markup " $label "] } lappend lines [join $refs $dot] return } proc EndIndex {} { upvar 1 lines lines lappend lines [markup

|  |  |
| --- | --- |
| }] lappend lines [markup "Keywords: $c"] lappend lines [markup | |

] # Engine parameter - insert 'footer' set footer [Get footer] if {$footer == {}} return lappend lines [markup ""] lappend lines [markup $footer] return } proc EndBody {} { upvar 1 lines lines lappend lines [markup ""] return } proc ALink {dst label} { markup " $label " } proc BeginColLeft {} { return [markup {}] } proc BeginColRight {} { return [markup {}] } proc SetAnchor {text} { return [markup " $text "] } proc Anchor {text} { global kwid cnt if {[info exists kwid($text)]} { return "\"$kwid($text)\"" } set anchor key$cnt incr cnt return "\"$anchor\"" } proc Row {} { global even set res [expr {$even ? "\#idxeven" : "\#idxodd"}] Flip return $res } proc Flip {} { global even set even [expr {1-$even}] return } # ### ### ### ######### ######### ######### ## Engine state # key : string -> dict(ref -> label) "key formatting" # ch : string -> '.' "key starting characters" # lk : string "last key" # la : string "index label" # ti : string "index title" # cnt : int # kwid : string -> ... # even : bool global key ; array set key {} global ch ; array set ch {} global lk ; set lk {} global la ; set la {} global ti ; set ti {} global cnt ; set cnt 0 global kwid ; array set kwid {} global even ; set even 1 global dot ; set dot [markup { · }] # ### ### ### ######### ######### ######### ## Engine parameters global \_\_var array set \_\_var { meta {} header {} footer {} kwid {} } proc Get {varname} {global \_\_var ; return $\_\_var($varname)} proc idx\_listvariables {} {global \_\_var ; return [array names \_\_var]} proc idx\_varset {varname text} { global \_\_var if {![info exists \_\_var($varname)]} {return -code error "Unknown engine variable \"$varname\""} set \_\_var($varname) $text return } ## # ### ### ### ######### ######### #########