Application Server in TCL/TK

An embedded, scripted, network-ready database framework to help with application development using TCL/TK

http://axn.dyndns.org

Table of Contents

1 Version history

0.0 Presentation at European TCL Conference 2009 in Strasbourg
0.1 changed to text format, added planned chapters

2 Notes on application deployment and maintenance

3 The case for an application server

4 A quick introduction into SQLite

- stable solution, was designed for TCL
- SQL syntax (if we should ever need to port)
- freedom of types like in TCL (eias principle)
- strings are longer than Oracle varchar2
- cross platform support like TCL
- can live purely in memory for webclients

5 The last TCL program we need

(insert as text listing!!!)

```
axn@w600:~$ more tcl/miniweb/dbappstarter.tcl
#!/usr/local/bin/wish

# ministarter from DB, working now ...

# binary lib, but included in most wrapped shells too
package require sqlite3
package require snit

# option to create DB and fill from web!!!

# using tclkit with sqlite inside, this could be deployed complete!!!
sqlite db [lindex $argv 0]

# proc to get a page from wiki
proc getpage {wiki page} {
        set code ""
        db eval {select value from tt where key = $page and wiki = $wiki} x {set code $x(value)}
        return $code
}

eval [getpage source main]
```

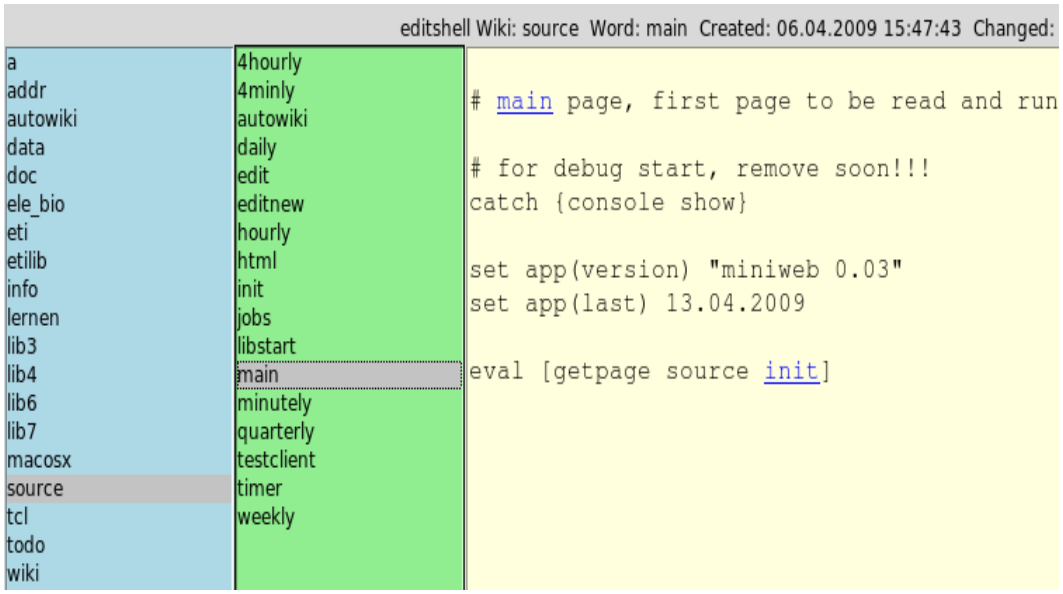
6 The world is a wiki

(insert as text!!!)

```
axn@w600:~$ sqlite3 tcl/miniweb/miniw600.dbapp
SQLite version 3.6.13
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select sql from sqlite_master where name = 'tt';
CREATE TABLE tt (id id, key string, value string, wiki string, first date, count number, last date)
sqlite> select distinct(wiki) from tt order by wiki;
a
addr
autowiki
data
doc
ele_bio
eti
etilib
info
lernen
lib3
lib4
lib6
lib7
macosx
source
tcl
todo
wiki
```

7 Code lives in wiki 'source'

(text!!!)



8 elements included

- simple editor page
- notebook page for multiple items in tabs
- timers for periodic jobs/refresh of pages
- updater
- bgerror and unknown catch and reporting
- library to import old wiki files, export HMTL
- client to update/refresh/merge remote wiki
- embedded http(s) server to deploy contentte of date last saved and save counter
- bgerror and unknown catch and reporting
- library to import old wiki files, export HMTL
- client to update/refresh/merge remote wiki
- embedded http(s) server to deploy content

9 To Do

- convert all pages from tool to loadable tabs
- define replication and synchronisation rules
- secure webserver and client with password
- run tabs in interp (Chrome does this …)
- deploy libraries and app as webmodules
- create templates for application types
- auto-update core app + libs via webserver

10 Examples of modules