FX – Fossil eXperimentally eXtended

Andreas Kupries
ActiveState Software Inc.
© 2014

21th Annual Tcl Conference
Embassy Suites Downtown, Portland, OR
Nov 10 – Nov 14, 2014

FX – Talking Points

- What (Is it)

- Why (Am I working on this)

- What (functionality does it have)

- Wanted (to have, and worked on)

- Where (to find)

FX – Overview

- Application (not a Package, nor Framework)
  - Despite using lots of packages for internal structure
  - Making them public in the future? Maybe!
- Provides functionality **fossil** does not have
  - And we found to or believe to need
  - Larger teams need management, and support for it
- Also a place for experimenting with things

FX – Motivations I

- **Situational awareness**
  - Easy with a small and/or dedicated Team
    - Daily Look at the Timeline (Commits, Ticket changes, …)
    - Communication easy (direct mail, talking with each other)
    - Most Tickets likely by the Team itself
    - Pull Model → Developers pull information from the repo
  - More difficult for (large(r)) OSS projects
    - (Lots) more changes coming through from the world
    - Team members have a Real Life™ → Intermittent Looks
    - Easy to miss things.
    - Push Model works better → Repo sends notifications.
  - Existing Fossil features do not cover the 2nd case

FX – Motivations II

- **Management** of related projects

  - Similar/same skins, similar/same ticket system

    – Meta-level development

  - And situational awareness again
    → **For the manager**

  - Current state

    – Most admin access must be done by web forms.

    – While CLI access to the config exists

      - Engineered for machine-readability, bulk transfer
        → Useful access effectively still web-only

        – Yet a text field is (IMHO) not a proper editor

    – Furthermore CLI access is coarse-grained
      → Read/write entire sections, not specific elements.

FX – Motivations III

- Result of the current state

    - Managing a family of related skins/ticket config is awkward at best to difficult at worst.

    - Easy to lose synchronization of config between the repositories

FX – Config Card Example

```
config /reportfmt 635
1379630989 'All Tickets' owner 'ticket-master' cols
'#ffffff Key:
#f2dcdc Open
#bde5d6 Deleted
#cacae5 Pending
#c8c8c8 Closed' sqlcode 'SELECT
CASE WHEN status=''Open''    THEN ''#f2dcdc''
     WHEN status=''Deleted'' THEN ''#bde5d6''
     WHEN status=''Pending'' THEN ''#cacae5''
     WHEN status=''Closed''  THEN ''#c8c8c8''
     ELSE ''#ffffff''
END AS ''bgcolor'',
rtrim(substr(tkt_uuid,1,10),''f'') AS ''#'',
datetime(tkt_mtime)   AS ''Last Changed'',
type                  AS ''Type'',
status                AS ''Status'',
subsystem             AS ''Subsystem'',
title                 AS ''Summary''
FROM ticket'
```

FX – Developer Awareness

- Change notifications via Email
  - Keep track of changes without notifications
    – Commits, Ticket changes, Wiki page changes, Attachments, and Corrections
  - **Fixed** destinations → Configured per type of change, for all types.
  - **Dynamic** destinations → Restricted to tickets.
    – Configure the ticket **fields** providing email addresses
  - Delivery triggered explicitly → **cron**, TH1/Tcl hooks
- Possible extension: Other transports
  - IRC (Subsume KGBbot ?), Tweets, …
- **Todo**: Ticket Moderation Notifications

FX – Notification Setup I

```
% fx note config
Default @ /home/www/fossil/tcl.fossil
+---+----------+--------------+---------------+
|   | Key      | Value        | Last-Changed  |
+---+----------+--------------+---------------+
| D | debug    | 0            |               |
| R | footer   | EuroTcl 2014 | Tue Sep 23 2  |
| D | header   | Automated ma |               |
| R | host     | mail.m4x.de  | Tue Sep 23 2  |
| D | limit    | 10           |               |
| R | location | http://core. | Tue Sep 23 2  |
| R | password | ************ | Tue Sep 23 2  |
| D | port     | 25           |               |
| R | sender   | akupries@sha | Tue Sep 23 2  |
| D | suspended | 0           |               |
| D | tls      | 0            |               |
| R | user     | tcl-commits@ | Tue Sep 23 2  |
+---+----------+--------------+---------------+
```

FX – Notification Setup II

```
% fx note routes
Default @ /home/www/fossil/tcl.fossil
+----------+------------------------------------+
| Event    | Route                              |
+----------+------------------------------------+
| commit   | tcl-bugs@lists.sourceforge.net     |
| control  | tcl-bugs@lists.sourceforge.net     |
| event    | tcl-bugs@lists.sourceforge.net     |
| ticket   | <assignee>                         |
|          | <closer>                           |
|          | <login>                            |
|          | <private_contact>                  |
|          | <submitter>                        |
|          | <username>                         |
|          | tcl-bugs@lists.sourceforge.net     |
| wiki     | tcl-bugs@lists.sourceforge.net     |
+----------+------------------------------------+
```

FX – Interna

- Where is the relevant information kept ?

  - Fossil repositories are sqlite3 databases

  - The namespace **'fx_%'** of table names is reserved for extensions and not touched by fossil itself

  - **Store in the repository – "**fx_aku_" prefix

- Main table → fx_aku_watch_seen

- 4 tables managing the destination ticket fields

  - fx_aku_watch_tktfield   – Chosen fields
  - fx_aku_watch_tktseen   – Processed tickets
  - fx_aku_watch_tkt   – UUID map of the tickets
  - fx_aku_watch_tktseries – Full history of the chosen fields.
  - Change can be in a destination → notify old and new

FX – Interna

- Test commands are available to inspect and debug all parts of the notification system separately
    - Manifest parsing
        - Rewritten twice
            - For syntax of signed commits.
            - For performance on encountering large tickets (Example: 400KB strace log)
    - Determination of mail destinations
    - Mail address parsing
    - Mail generation
    - Mail setup

FX – Management Awareness I

- Mostly **To Come**

- Focus on Ticket system

- Some foundations

  - User-defined enumerations (create, delete, modify)

  - User-defined dictionaries (aka mappings) (ditto)

FX – Management Awareness I

- Mostly **To Come**; Focus on Ticket system

- Some foundations

  - User-defined enumerations (create, delete, modify)
  - User-defined dictionaries (aka mappings) (ditto)

- Ticket system changes are a hassle

  - Example: Lists of Categories, Users
    - **Current:** Specified in TH1 code. Must edit code to change (Then replicate N times).
    - **Instead**: Link to enumerations and dictionaries
      - Code editing only once, at link time. Automatic.
    - Easy CLI access after, no code changes.

FX – Enum/Dict Setup

```
set subsystem_choices {
    {- New Builtin Commands}
    {- New Builtin Object Types}
    {- New Stub Entries}
    {01. Notifier}
    {02. Event Loops}
...
}


set subsystem_choices [query {
    SELECT item FROM fx_aku_enum_subsystem
    ORDER BY id
}]
```

FX – Enum/Dict Setup

- Note: Users are special.

    - No access to their table from TH1 code
      → **Good**: Unable to expose password (hashes)
      → Yet also makes us unable to pull a simple user list into ticket forms without "manual management" ≡ editing TH1 code

    - External access is still possible.

    - Currently thinking about setting up and using a
      → **shadow enum**

        – Would be updated explicitly (cron).

        – And linkable like a regular enum.

FX – Features for Mgmt II

- Mostly **To Come**, focus on Ticket system
  - Direct editing of
    - HTML/TH1 ticket forms, reports, ticket db tables
    - Fine-grained access to the other parts of the setup
  - Still working out the CLI API
  - **Wants**
    - Parameterized Reports – CLI first, maybe Web?
    - Rule/Action system
      - Ticket auto-assignment (via category information)
      - (In)Activity warnings
      - Spam detection?
    - Searchable Tickets, Commits, Code

FX – Other Features

- Peering
    - Management of a web of repositories
        - Redundancy, Backup
    - Multi repository synchronization
        - Action chosen per destination
    - Git Mirror (push-only)

FX/Fossil – More Ideas

- Get notification on new (self-registered users)
- Get notification for moderated ticket changes

  - Have embedded links for approve/delete ops?

- Forgotten password web-form ?
- Ticket moderation Page – Uplink to mod_list
- Restrict cli timeline to a branch, around a date
- Supporting code

  - Bash completion definition

FX/Fossil – Yet More Ideas

- Official Repositories to collect
  - User-contributed skins/themes
  - User contributed Ticket system configs.

FX – Location

Where ?

- The Conference eProceedings ($\equiv$ USB-Stick)

- http://core.tcl.tk/akupries/fx

- http://chiselapp/user/andreas_kupries/repository/fx

- Related projects

  - fcommit – Tk GUI – http://wiki.tcl.tk/fcommit

FX – References

Supporting packages & tools

- The Conference eProceedings ($\equiv$ USB-Stick)

- http://core.tcl.tk/akupries/cmdr

- http://core.tcl.tk/akupries/linenoise-utilities

- https://github.com/andreas-kupries/tcl-linenoise

  - https://github.com/andreas-kupries/linenoise

  - https://github.com/msteveb/linenoise

    - (https://github.com/antirez/linenoise)

- https://core.tcl.tk/akupries/kettle

FX – Fossil eXperimentally eXtended

Thank you for listening

Questions ?

Or talk to me later