1 Managing Data with Version Control

This repo goes along with a presentation given at MacTech Conference 2011

Figure 1: MacTech Conference 2011

1.1 Notes about my particular process

I create new files in “notes”, which are transformed into the XML worklog syntax, which is stored in “worklogs”. This syntax is used to create PDF invoices in “invoices” or ledger files in “ledgers”.

I don’t include the worklog parsing/ledger/pdf generation code itself because it’s pretty ancient, in PHP, has a bunch of weird non-current library dependencies and does some really stupid things (for example, storing unix timestamps in the XML, rather than ISO–8601 dates).

The rest of the files in the repo were added after the fact (like this README which you are reading right now).

1.2 Tools

1.2.1 Automation Tools

- Git
  - Documentation and tutorials
    - `man gittutorial`
    - Everyday Git
    - Pro Git
    - Git for Computer Scientists
- Mercurial
  - Documentation and tutorials
    - Hg Init
    - Mercurial: The Definitive Guide
- Rake
  - A tool for writing dependency based tasks, in Ruby
  - Tutorials
    - Using the Rake Build Language
    - Using Rake to Automate Tasks

1.2.2 Documentation Tools

- Markdown
  - minimal markup language, widely implemented
  - One example: MultiMarkdown
- Pandoc
  - Documentation format conversion swiss army knife
  - Will convert Markdown to HTML, ePub, PDF, etc.

1.2.3 Accounting Tools

- Ledger
  - Ledger is a CLI accounting tool
  - Multiple implementations
    * the original C++ version, given above
    * Beancount in python
    * hledger in haskell
  - The 3.x beta branch off of github is preferred over the older 2.6.x version in MacPorts

1.3 Scripts

Note that these are pulled straight from my code, and won't work without fairly serious modifications - they often will need other utilities installed. Use this as a starting point

1.3.1 `scripts/docgen`

This is the documentation generation script that I use to convert markdown files into different formats - for example, turning this README into a nicely formatted PDF and HTML docs that are also in the repo.

It's dependent on pandoc, imagemagick, pdfcrush, advpng and a few other XML specific tools like jing.

See the `test` subfolder for examples - all the files inside were created from `testfile.mkd`

1.3.2 `scripts/mailsend.scpt`

A very simple Applescript that will send an email with specified subject/sender/attachments. I run this with a bash script that calls rake. The code as written below needs help to be functional in your environment.

bash script:

```
#!/usr/bin/env rake
RAKECMD=”rake -f ~/tools/worklog.rake”
${RAKECMD} ”${1}” target=”${2}”
```

rakefile - at ∼/tools/worklog.rake in my environment

```
# paths
tool_path = ”~/tools”

# binaries
$osascript_bin = ”/usr/bin/osascript”

# scripts
$mailsend_scpt = tool_path + ”/worklog/mailsend.scpt”

# set the target variable if it’s a valid file
$target = nil
if !ENV[”target”].nil? && File.file?(ENV[”target”])
    $target = ENV[”target”];
end

# mailinvoice
desc ”Create an email message with attachment in Mail.app”
task :mailinvoice => [ :worklogs, :ledgers ] do
    if $target
        note_date = ‘date +”%Y-%m-%d”‘
        filepath = File.expand_path($target)
        name = customer_name()
        email = customer_email()
     sh ”#{$osascript_bin} #{$mailsend_scpt} ’Invoice for #{note_date}’ ’Please see the enclosed invoice\
    end
end

def customer_name()
    return ”customer”
end

def customer_email()
    return ”user@host.net”
end
```