

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 “invoice” or ledger files in “ledgers”. I don’t include this code because it’s pretty ancient, 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 Scripts

Note that none of these are fully prepped to work on any system - they often will need other utilities installed, or will need to be fixed to work in your specific environment. These are mainly starting points.

1.2.1 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`

1.2.2 scripts/mailemail.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:

bash script:

```
$FOCUSTOOLPATH=~/.tools
RAKECMD="rake -f ${FOCUSTOOLPATH}/worklog/worklog.rake"
${RAKECMD} "${1}" target="${2}"
```

rakefile:

```
# 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
task :mailinvoice => [ :worklogs, :ledgers ] do
  if $target
    note_date = ENV['FOCUSDATE'].gsub('_', '-')
    filepath = File.expand_path($target)
    name = customer_name()
    email = customer_email()
    sh "#{$osascript_bin} #{mailsend_scpt} 'Invoice for #{note_date}' 'Please see the enclosed invoice\
  end
end
```

1.3 Tools

1.3.1 Automation Tools

- [Git](#)
 - A version control system
 - Documentation and tutorials
 - * [man gittutorial](#)
 - * [Everyday Git](#)
 - * [Pro Git](#)
 - * [Git for Computer Scientists](#)
- [Rake](#)
 - A tool for writing dependency based tools, in Ruby
 - Tutorials
 - * [Using the Rake Build Language](#)

1.3.2 Documentation Tools

- [Markdown](#)
 - minimalist markup language
 - Multiple implementations, for example [MultiMarkdown](#)
- [Pandoc](#)

- Documentation format conversion swiss army knife
- Will convert Markdown to HTML, ePub, PDF, etc.

1.3.3 Accounting Tools

- [Ledger](#)
 - Ledger is a CLI accounting tool
 - Use the 3.x beta branch off of github, which is being actively developed