Another System Definition Facility version 3.1

A traverse across the build

A monster hunt story

François-René Rideau <tunes@google.com>
Based on a true story!
Based on a true story!

Setting: a seemingly peaceful little village
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter

Order: New sheriff in town
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter

Order: New sheriff in town

Nemesis: a monster that terrorizes innocent people
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter

Order: New sheriff in town

Nemesis: a monster that terrorizes innocent people

Fight: not easy as thought, but long and hard
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter

Order: New sheriff in town

Nemesis: a monster that terrorizes innocent people

Fight: not easy as thought, but long and hard

Tension: climaxes! twists! magic help!
Based on a true story!

Setting: a seemingly peaceful little village

Protagonist: young and naïve new hunter

Order: New sheriff in town

Nemesis: a monster that terrorizes innocent people

Fight: not easy as thought, but long and hard

Tension: climaxes! twists! magic help!

Conclusion: respect acquired and lessons learned
The setting: ASDF

- build system
- components
- operations
- action graph
- plan
- in-memory
Defining components

(defsystem "fare-quasiquote" ...
  :depends-on ("fare-utils")
  :components
    ((:file "packages")
     (:file "quasiquote"
      :depends-on ("packages"))
     (:file "pp-quasiquote"
      :depends-on ("quasiquote"))))
(defsystem "fare-utils" ... :components
  ((:file "package")
   (:module "base" :depends-on ("package")
    :components
     ((:file "utils")
      (:file "strings" :depends-on ("utils")
       ...)))
   (:module "filesystem" :depends-on ("base")
    :components ...
    ...
    ...))
)

ASDF: summary

- component = system | module | file ...
- a component depends-on other components
- operation = load-op | compile-op | ...
- operations propagate along the component DAG
- action = operation x component
- an action depends-on other actions
- action DAG ≠ component DAG
- plan = topologically sorted list of actions
Also in the extended article...

CL is Now an Acceptable Scripting Language

Why it rocks / sucks compared with C build tools

Innovations in ASDF 1 2 2.26 3 3.1

The Problem with Pathnames

Lessons in Software Design including Pitfalls

http://github.com/fare/asdf3-2013
Use it!

http://common-lisp.net/project/asdf/

http://github.com/fare/asdf3-2013

Any Questions?