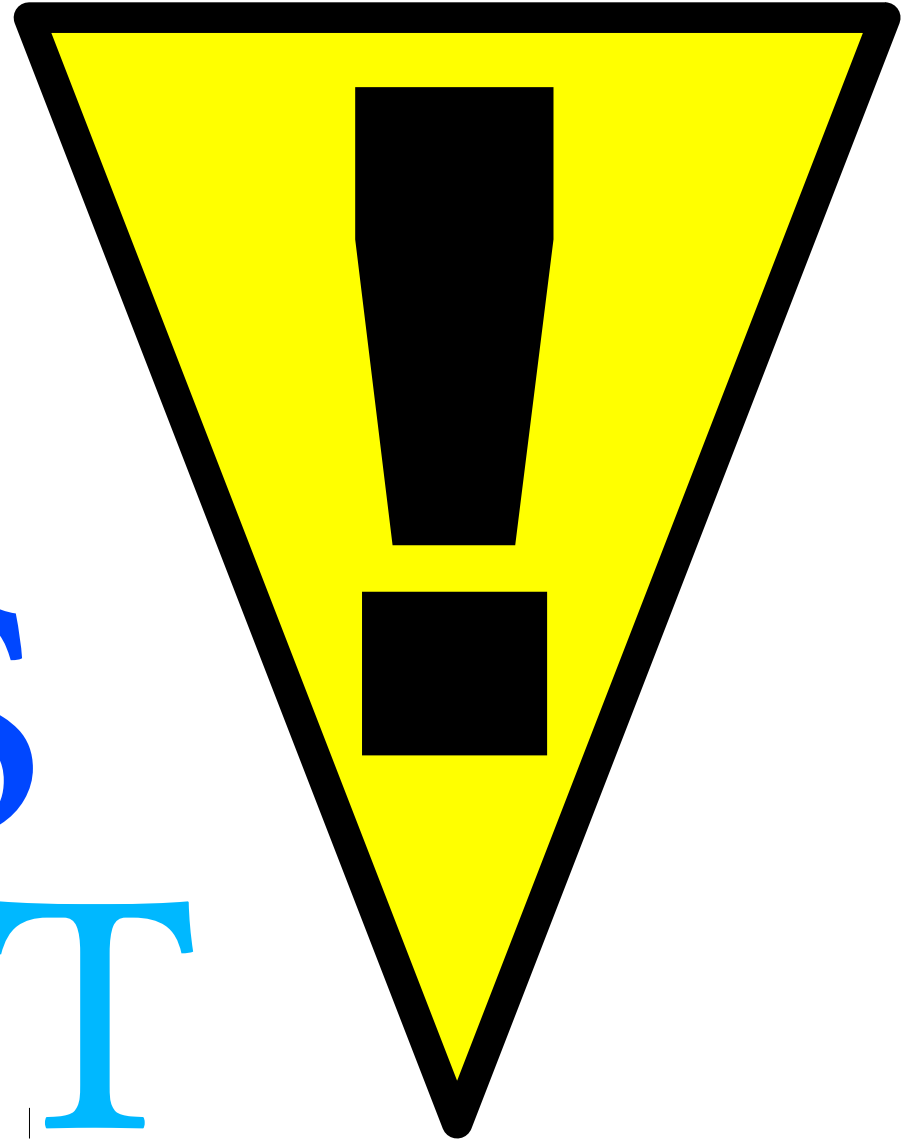




ACL2

BOOKS

REPORT





# COMMUNITY'S WORK

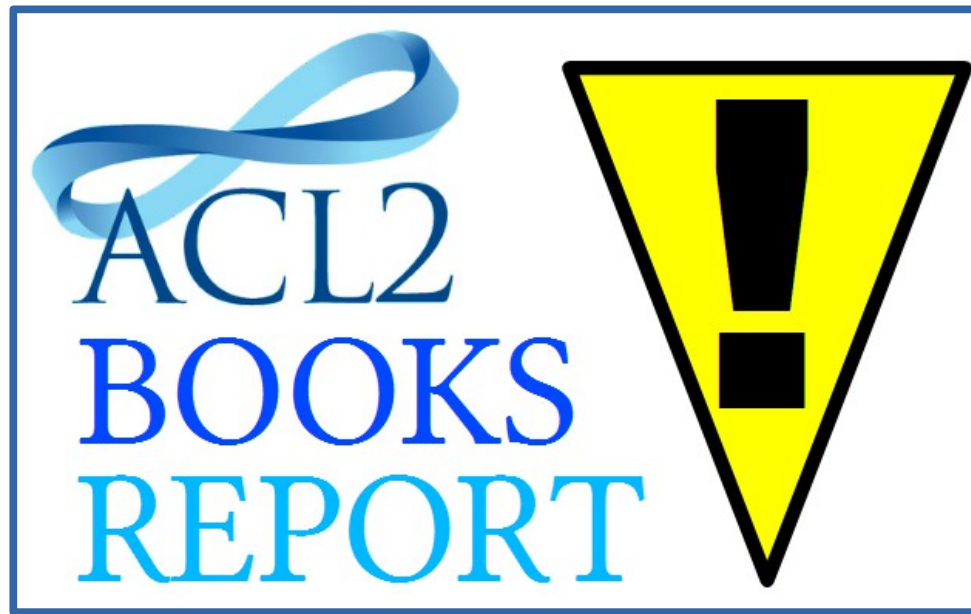
(I'm just a reporter)

# REPORT

~~Jared Davis~~

~~Central Technology~~

ACL2 Workshop 2015



# Rough Scope

Changes from ACL2 6.3-7.1  
(October 2013-May 2015)

NOTE-{6-4,6-5,7-0,7-1}-BOOKS

# Current Status

# 2.7 million lines of Lisp

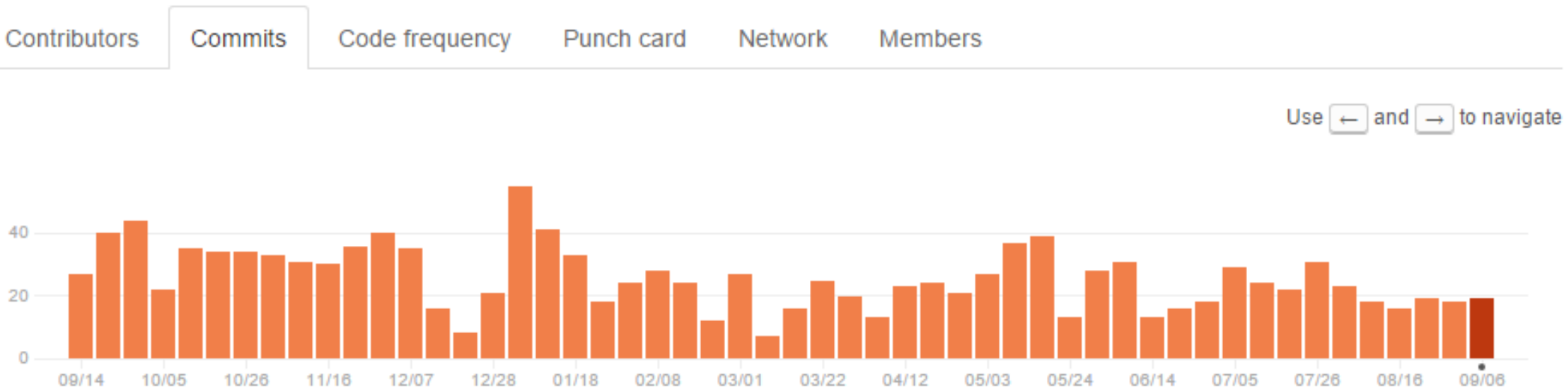


6,450 sources



1,000 dirs

 x 10



~20 commits per week

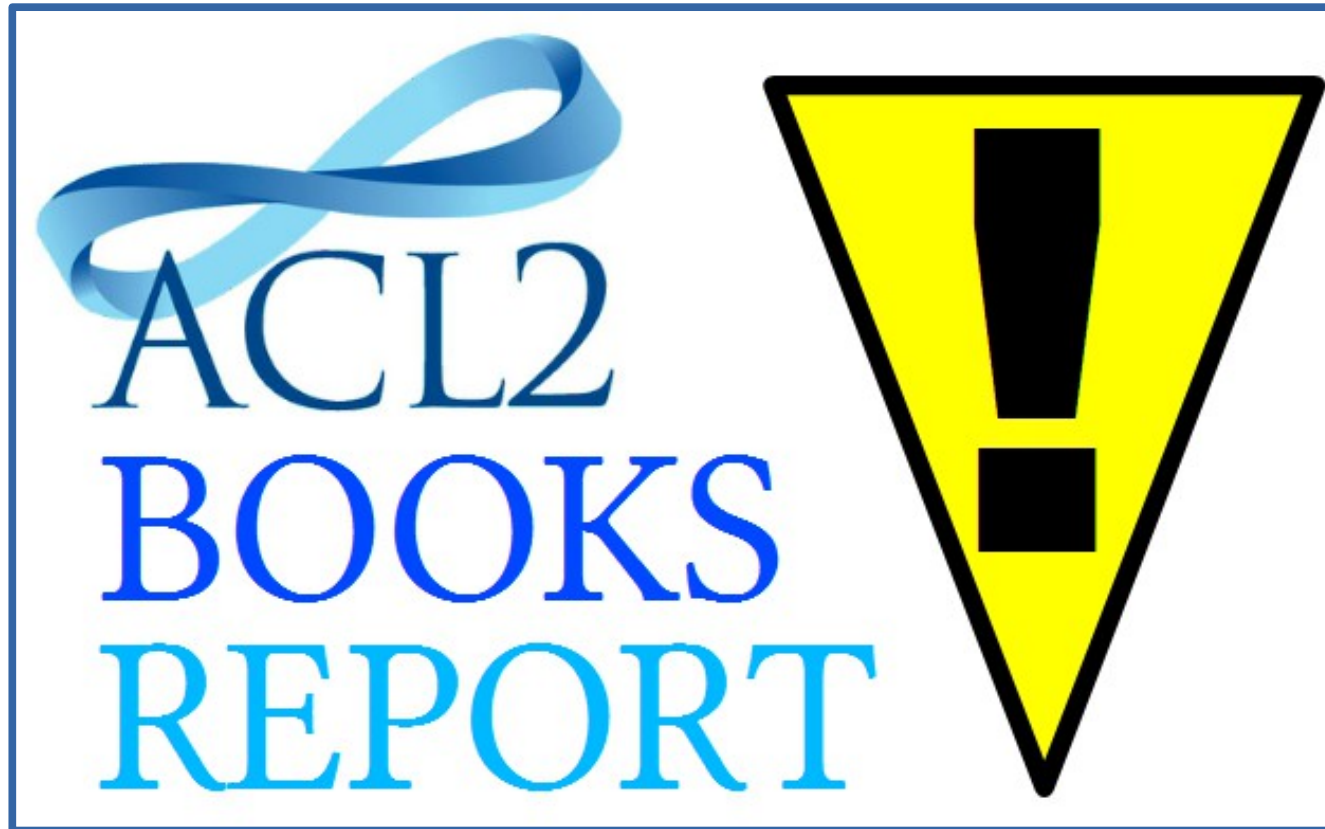
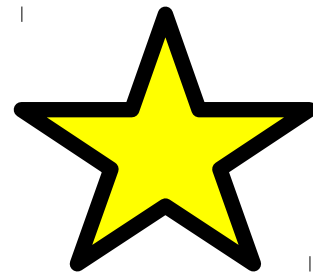
so far in 2015:

5,726 files changed

+1M lines

-245K lines

How to keep up?



# Note-6-4-books

[books]/doc/relnotes.lisp

Release notes for the ACL2 Community Books for ACL2 6.4 (January, 2013).

The following  
ACL2 6.3 and  
numbers co

For addition

**Build S**

In previous  
take several  
**books.**

This partic  
changed ho  
(nearly) all

6.4.1



Jump to

append

Search

files

Note-6-5

# Note-6-5-books

[books]/doc/relnotes.lisp

Release notes for the ACL2 Community Books for ACL2 6.5 (August 2014).

The follow  
ACL2 6.5

The acl2  
correspo  
details, y

**Org**

**Dele**

When w



Jump to

append

Search

files

Note-7-0

# Note-7-0-books

[books]/doc/relnotes.lisp

Release notes for the ACL2 Community Books for ACL2 7.0 (January 2015)

The follow  
ACL2 6.5

The acl2-  
correspon  
details, yo

**Org**



Jump to

append

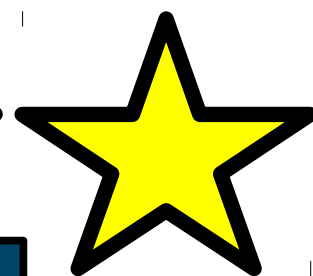
Search

files

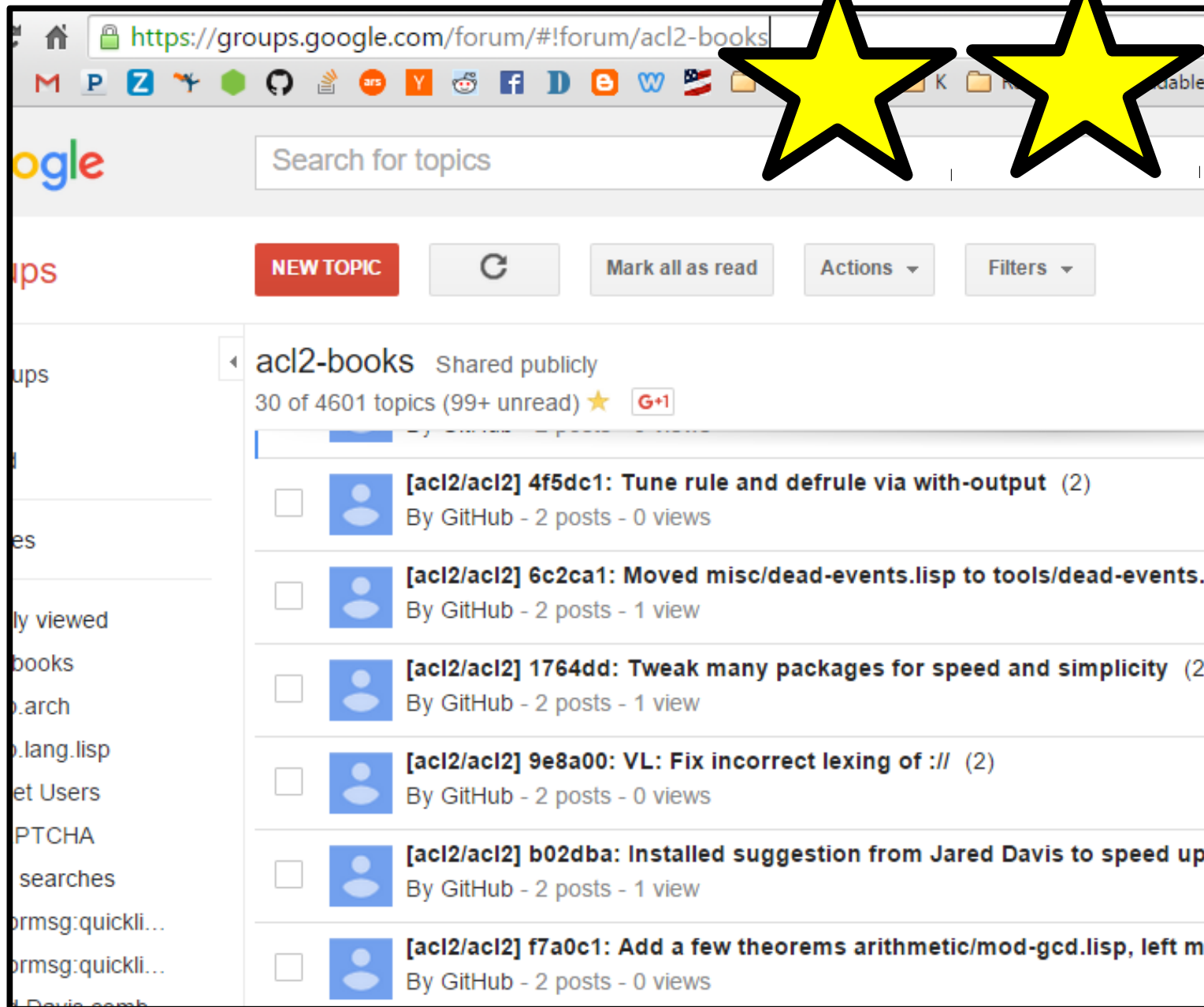
Note-7-1

# Note-7-1-books

[books]/doc/relnotes.lisp







The image shows a screenshot of a Google Groups forum page for 'acl2-books'. The browser's address bar shows the URL 'https://groups.google.com/forum/#!forum/acl2-books'. The page features a search bar, a 'NEW TOPIC' button, and navigation options like 'Mark all as read', 'Actions', and 'Filters'. The forum title is 'acl2-books Shared publicly' with '30 of 4601 topics (99+ unread)'. A list of topics is displayed, each with a checkbox, a user icon, a title, and a 'By GitHub' attribution with post and view counts.

Topic Title	Author	Posts	Views
[acl2/acl2] 4f5dc1: Tune rule and defrule via with-output (2)	GitHub	2 posts	0 views
[acl2/acl2] 6c2ca1: Moved misc/dead-events.lisp to tools/dead-events.	GitHub	2 posts	1 view
[acl2/acl2] 1764dd: Tweak many packages for speed and simplicity (2)	GitHub	2 posts	1 view
[acl2/acl2] 9e8a00: VL: Fix incorrect lexing of :// (2)	GitHub	2 posts	0 views
[acl2/acl2] b02dba: Installed suggestion from Jared Davis to speed up	GitHub	2 posts	1 view
[acl2/acl2] f7a0c1: Add a few theorems arithmetic/mod-gcd.lisp, left m	GitHub	2 posts	0 views

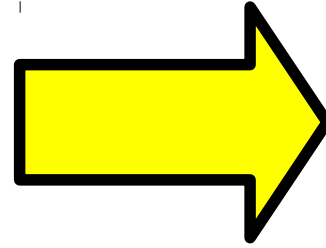
# ACL2-Books Google Group

The image shows a screenshot of a GitHub repository page for 'acl2 / acl2' and a gitk window. The GitHub page displays the repository name, a search bar, and a list of commits. The gitk window shows a commit graph with branches like 'master', 'remotes/origin/master', and 'remotes/origin/testing'. A diff view is open for a commit with SHA1 ID '62eb9ec514f5cd0b360d58d860dfb5845d78b796', showing changes to 'books/centaur/sv/svtv/doc.lisp'. The diff includes comments about XML generation and documentation. Four yellow stars are overlaid on the top of the screenshot.

GitHub, gitk, etc.

GitHub

Google code



**Unified ACL2+Books**

[github.com/acl2](https://github.com/acl2)

**Fork me on GitHub**



# Build System

# Don't build so much!

cd books

make -j 4

## Basic Build

arithmetic

std

arithmetic-3

tools

arithmetic-5

misc

ihs

xdoc

data-structures



# Customize your build

```
#!/bin/sh
```

```
make -j 4 basic \  
  ordinals \  
  centaur/gl/gl.cert \  
  coi/bags/top.cert \  
  ...
```



# More info

Community-books

## Books-certification

ACL2 Sources

Instructions for certifying the ACL2 [community-books](#).

Starting in ACL2 6.4 we recommend using the new Community Books `make` system to certify the books. If you encounter problems using the new system (described below) or need some feature that is no longer available, please see [books-certification-alt](#) for alternate instructions. We have *not* changed how `make regression` works from the `acl2-sources` directory.

## Prerequisites

We assume that you have already downloaded and installed ACL2 as per the [ACL2 installation instructions](#) on the ACL2 home page.

# Boring stuff

Bugs fixed

Reorganization

Portability improved

ACL2(r) build unified

Dependencies reduced

Slow books sped up (accumulated persistence)

Hundreds of Makefiles deleted



# Ooh, Colors!!!



```
logic 200575 Sep 21 15:05 lvalues.cert  
/fv/jared/xc/ac12/books/centaur/sv/svex/compose.cert on 21-Sep-2015 15:  
fv/jared/xc/ac12/books/centaur/sv/svex/compose.cert (6s)  
logic 176684 Sep 21 15:05 compose.cert  
/fv/jared/xc/ac12/books/centaur/sv/v1/svstmt-compile.cert on 21-Sep-2015  
fv/jared/xc/ac12/books/centaur/v1/loader/parser/expressions.cert (57s)  
logic 636343 Sep 21 15:05 expressions.cert  
/fv/jared/xc/ac12/books/centaur/v1/loader/parser/datatypes.cert on 21-Sep-2015  
/fv/jared/xc/ac12/books/centaur/v1/loader/parser/lvalues.cert on 21-Sep-2015  
fv/jared/xc/ac12/books/centaur/sv/v1/svstmt-compile.cert (13s)  
logic 192841 Sep 21 15:06 svstmt-compile.cert  
fv/jared/xc/ac12/books/centaur/v1/loader/parser/lvalues.cert (8s)  
logic 60086 Sep 21 15:06 lvalues.cert  
fv/jared/xc/ac12/books/centaur/v1/parsetree.cert (93s)  
logic 7975852 Sep 21 15:06 parsetree.cert
```

```
export CERT_PL_NO_COLOR=1
```

# Build Documentation

Books

## Cert.pl

[books]/build/doc.lisp

`cert.pl` is a mature, user-friendly, industrial-strength tool for certifying ACL2 books.

### Introduction

For "pure" ACL2 projects—even large ones—`cert.pl` will let you to certify any book, whenever you like, without writing any Makefiles. If your book includes supporting books that aren't certified, `cert.pl` will rebuild exactly the necessary books, in parallel, even if they're in other directories.

# Build Tools

```
$ critpath.pl centaur/v1/kit/top.cert
```

```
Critical Path
```

File	Cumulative	Time	Speedup
kit/top.cert	12.2 min	52.1 sec	52.1 sec
kit/lint.cert	11.4 min	1.0 min	1.0 min
unparam/top.cert	10.3 min	36.9 sec	4.8 sec
unparam/override.cert	9.7 min	22.5 sec	4.8 sec
v1/elaborate.cert	9.3 min	2.9 min	2.2 min
v1/v1-svstmt.cert	6.4 min	36.0 sec	36.0 sec
v1/expr.cert	5.8 min	1.0 min	44.9 sec
mllib/selfsize.cert	4.8 min	20.2 sec	4.2 sec
mllib/hid-tools.cert	4.5 min	47.7 sec	47.7 sec
mllib/scopestack.cert	3.7 min	42.9 sec	42.9 sec
mllib/blocks.cert	2.9 min	24.1 sec	5.1 sec
v1/parsetree.cert	2.5 min	1.5 min	1.5 min
v1/expr.cert	1.0 min	36.9 sec	8.3 sec
bitops/ihsext-basics.cert	24.2 sec	11.3 sec	4.8 sec

# Build Tools

`clean.pl` – run anywhere to recursively delete certs, compiled files, and so on

`memsum.pl` – summarize gc messages from `.cert.out` files (ccl only)

# Name Conflicts

# Many Name Clashes Fixed

Arithmetic-2/3/5 vs IHS

Bitops vs RTL (sign-extend)

Bitops vs Arithmetic-5

COI vs CCG (remove-keywords)

COI vs data-structures (package)

COI vs Std/osets (deep)

COI vs Std/lists (repeat)

COI vs Witness-CP



# Working-with-packages

[books]/doc/practices.lisp

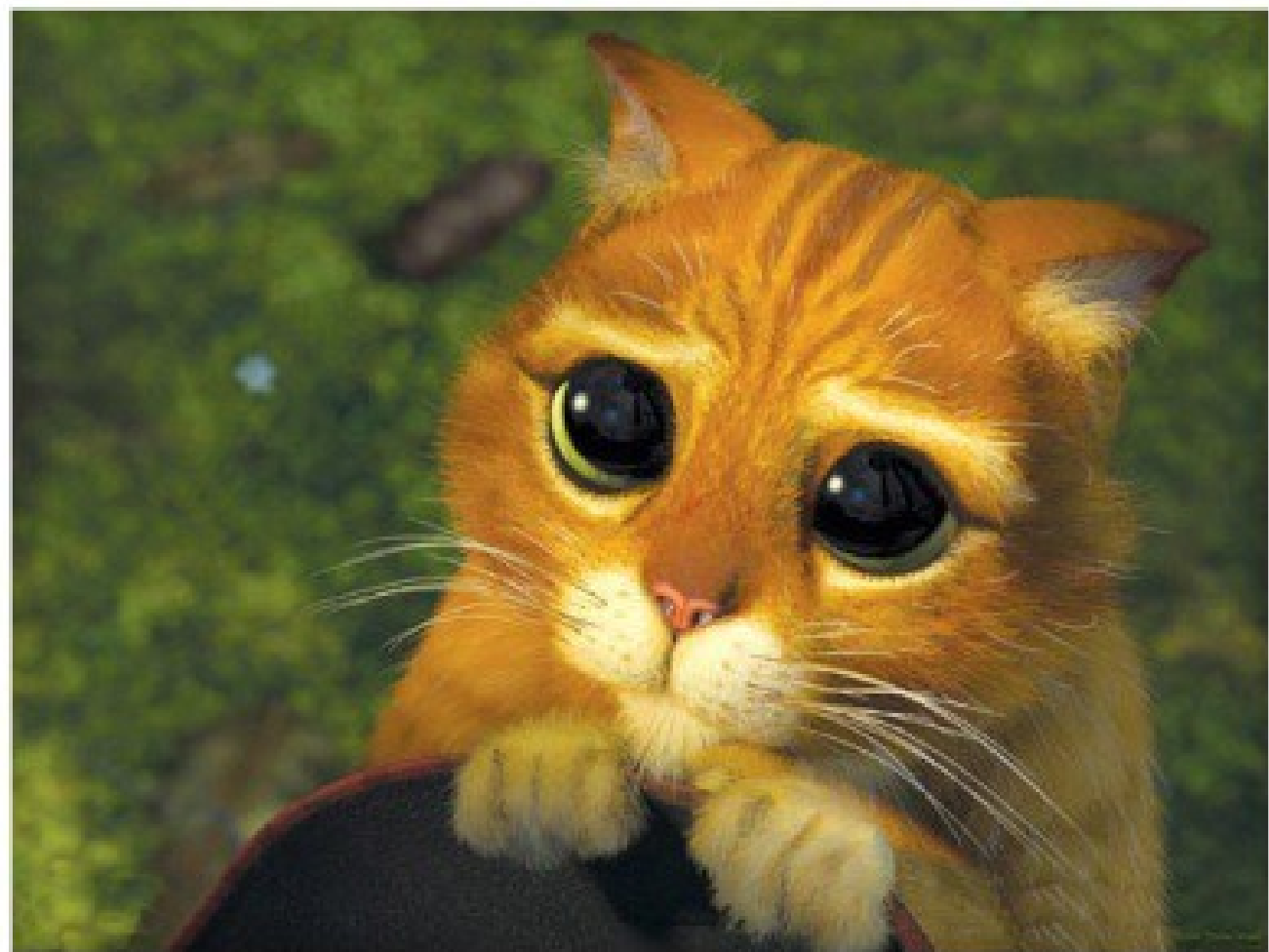
How to set up new package and portcullis files.

## Recommendations

Here is a basic recipe to follow for creating packages:

**foo/package.lisp** — main package

```
;; load other packages needed
(include-book "lib1/portcullis")
```





YOU SHALL NOT PASS!

Standard

# Std/Basic (new)

- Induction schemes
- Basic fixing functions and equivalences

## Lists, Alists, Osets, Bitsets

- Minor updates

# Std/Util

Define – Awesome defun replacement

Defines – define for mutual recursion!!  
with make-flag integration!!!

Defret and ret b\* binder

Defaggregate/Deflist/Defalist/etc.

Defval – defconst with xdoc support

Defsum – primitive sum of aggregates

Defaggrify-defrec – for ACL2 hackers

Defredundant – like it sounds

# Std/Util

Define-like syntax throughout

Smoothing out the rough edges

With-output

XDOC integration

Performance tuning

Compatibility with FTY

# Std/Strings

- **Pretty** – logic mode pretty printer
- Rich **numeric functions**
- **Base64** encoding/decoding
- Program mode book
- Simplified logical definitions
- Use define for better docs

# Std/IO

- **Read-string:** parse an s-expression from a string
- Print-legibly, print-compressed

## Std/typed-lists (new)

- For integer-listp, etc.



Xdoc

# New Documentation

arithmetic-1

arith-equivs

best-practices

bitops

base64

cert.pl

cowles

clause-processor-tools

defconsts

defsort

defrstobj

def-universal-equiv

ihs

projects/\*

rtl

set-max-mem

sneaky

plev

# Features

Legacy docs converted

Topic hierarchy improvements

Browser speed, compatibility

Mobile-friendly edition (yaya!)

Link checking, error reporting

Search engine optimization

# Quicklisp and OSLIB

# Quicklisp

Zach Beane

bordeaux-threads

cl-fad (pathnames)

osicat (filesystem & misc os stuff)

uiop (filesystem stuff)

shellpool (subprograms)

hunchentoot (web server)

...

## Quicklisp beta

Quicklisp is a library manager for Common Lisp. It works with your existing Common Lisp implementation to download, install, and load any of over 1,200 libraries with a few simple commands.

Quicklisp is easy to install and works with ABCL, Allegro CL, Clasp, Clozure CL, CLISP, CMUCL, ECL, LispWorks, MKCL, SBCL, and Sciener CL, on Linux, Mac OS X, and Windows. The libraries were last updated on September 24, 2015.

To get started with the Quicklisp beta, download and load  
<https://beta.quicklisp.org/quicklisp.lisp>

# Quicklisp Build

```
cd books
```

```
make USE_QUICKLISP=1 . . .
```

## New Bundle System

No extra downloads

Everything is just in Git

# OSLIB

File stuff like `ls`, `cp`, `rm`, `mkdir`

File type querying like `test -d`

Misc Path and other OS stuff

Command line arguments

(see also `getopt`)

# Quicklisp

[books]/centaur/quicklisp/top.lisp

An ACL2 connection to the [Quicklisp](#) system for installing Lisp libraries.

## About Quicklisp

Quicklisp is a Common Lisp library that can be installed via [RubyGems](#) for [Ruby](#). It is the basis for many of Common Lisp libraries.

To make it easy to use Quicklisp, we have placed the books in the books/centaur directory. These books are the basis for other libraries. Some of these books require [trust-tag](#).

ACL2::interfacing-tools

# Oslib

[books]/oslib/top-logic.lisp

Operating System Utilities Library

This is a collection of ACL2 functions that allow you to do various basic operating-system related tasks, e.g., you can get the current PID or user name, file listings, etc.

Almost everything here necessarily requires a trust tag, because it is implemented in raw Lisp. We believe we have connected this functionality to ACL2 in a sound way, using [read-ACL2-oracle](#).

The library is far from complete since we tend to extend it only as the need

OSLIB  
Package



# Getopt

[books]/centaur/getopt/top.lisp

GETOPT  
Package

A library for processing command-line option.

## Introduction

**Getopt** is a tool for writing command-line programs in ACL2. It is similar in spirit to [Getopt::Long](#) for Perl, [Trollop](#) for Ruby, and so on.

We basically extend [defaggregate](#) with a command-line parsing layer. This has some nice consequences:

- Argument parsing gives you an ordinary aggregate that can have nice, strong type guarantees on its fields.

# Arithmetic

# RTL

Deleted old versions

Moved into a package

GL integration

And ... ???

# Bitops

Now in a package

Many new fast functions

Scattered theorem improvements

Much more documentation

# Hardware Verification

Stable

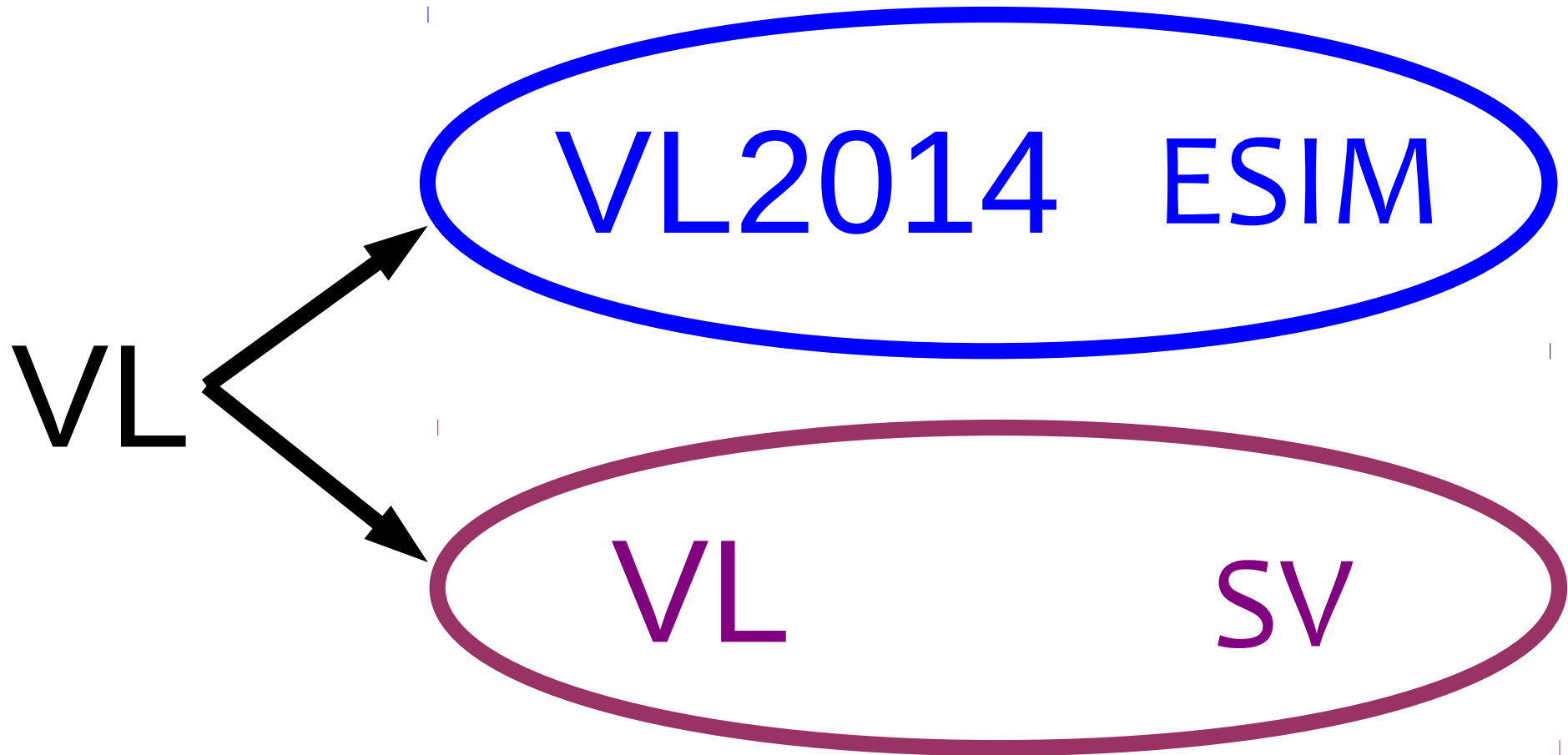
VL2014 ESIM

VL

VL

SV

Under Development  
Greatly Expanded



Many GL optimizations

Improved 4v-sexpr/SV rewriting

SAT solvers now easily pluggable

Scripts for SAT proof checking

Cross-Lisp portability (e.g., tshell)

# New Libraries and Tools



## **tools/last-theory-change**

See when a rule was last enabled/disabled

## **centaur/misc/dag-measure**

A measure for algorithms over acyclic graphs

## **misc/enumerate**

A trick for breaking a proof into many cases

## **remove-hyps**

Identify unnecessary hyps in your theorems

## **tools/rewrite-with-equality**

Aggressively use equality hyps in stable goals

## **with-supporters and defredundant**

Automatically produce redundant events

## **flag::def-doublevar-induction**

Prove congruences about mutual recursions

## **centaur/nrev**

An alternative to nreverse in ACL2

## **projects/sidekick**

A graphical add-on for ACL2

## **system/toothbrush**

Create smaller ACL2-based applications

## **depgraph**

Generic graph algorithms (topological sort, transitive dependencies, inversion, ...)

## **projects/codewalker**

## **projects/hybrid-systems**

Shant Harutuntian's PhD

## **spacewalk**

Understand heap memory usage (CCL only)

## **simp**

Ask ACL2 to simplify a term under certain hyps

## **misc/check-fn-inst**

Check constraints to a functional instantiation

## **def-saved-obligs**

Save proof obligations as separate defthms

## **drat-trim**

Check SAT solver proofs

## **clause-processors/induction**

A clause processor that does induction

## **numerous new demos and examples**

# Licensing



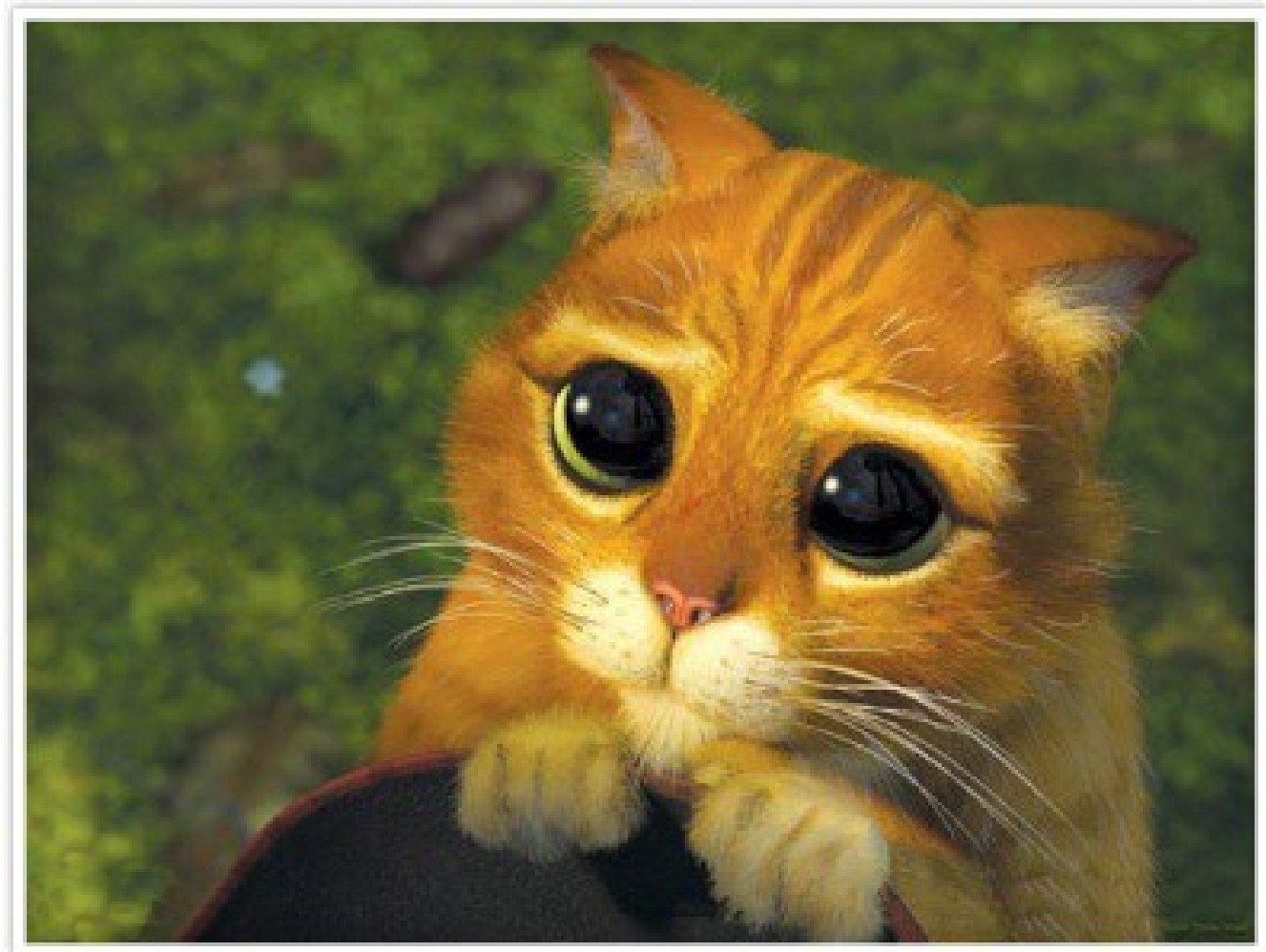
AMD & Intel (RTL)  
Centaur & Friends  
Computational Logic  
COI Books  
Krug (Arithmetic)  
Northeastern (Ordinals)  
Oracle  
Rager  
Selfridge  
UT Austin

# Omissions



7.1 onwards  
x86 books(!)  
cgen/defdata  
fty, smtlink, etc.  
ACL2(r) stuff  
demos  
probably lots more

# Report Any Problems!





Thanks  
Everyone  
(Now get back to work!)