Workshop: High-performance computing for economists

Lars Vilhuber[1] John M. Abowd[1] Richard Mansfield[1]
Kevin L. McKinney

[1]Cornell University, Economics Department,

August 20-22, 2013: Day 2

What are VCS

Software development perspective

Version control systems (VCS) (or Source Configuration Management (SCM) systems $*$) allow developers or authors to keep track of the history of their projects? source code. [source]

What are VCS

Software development perspective

Version control systems (VCS) (or Source Configuration Management (SCM) systems $*$) allow developers or authors to keep track of the history of their projects? source code. [source]

Generic view

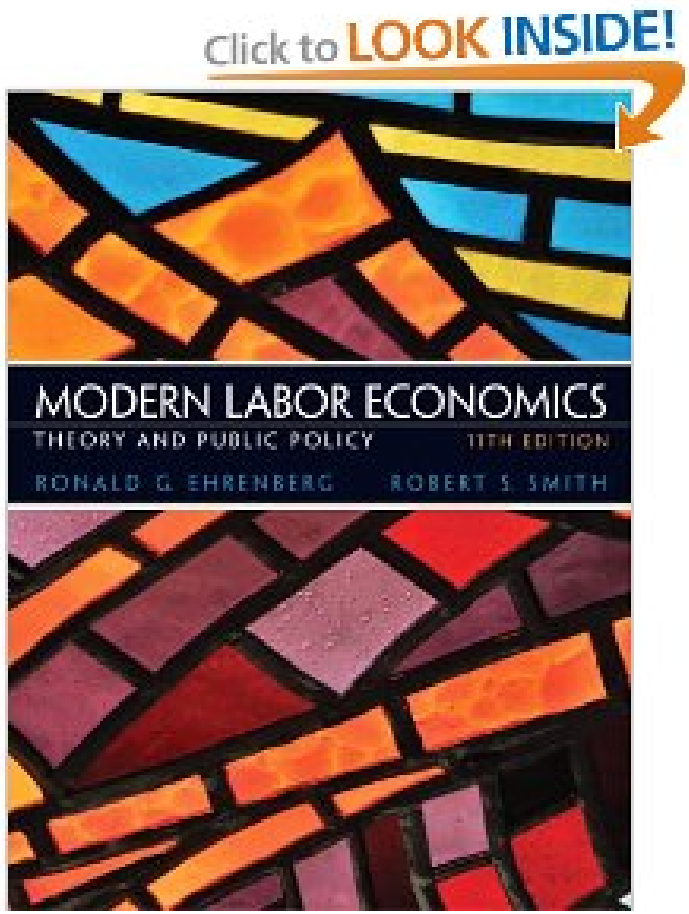
Detailed mechanism to manage different versions (historical, parallel) of documents, files, programs, etc.

You already have it

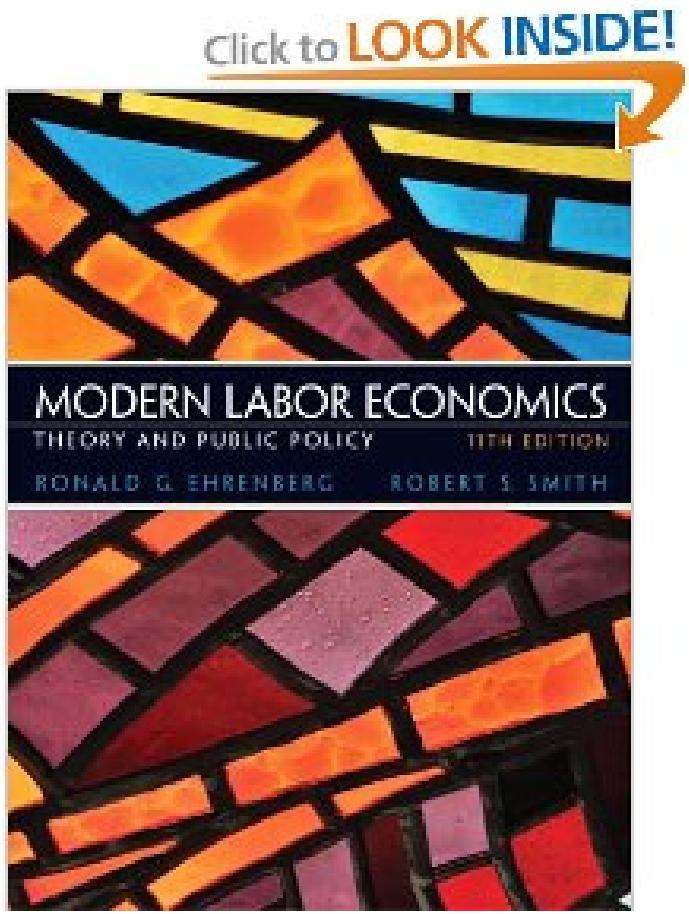
Implicit uses

- Backup systems (Apple Time Machine, others)
- Word processors (Undo feature, Track changes in Word, finer-grained features in content management systems: Google Docs, Blog software, etc.)
- Versioning filesystems
- Paper books!

Modern Labor Economics: Theory and Public Policy



Modern Labor Economics: Theory and Public Policy

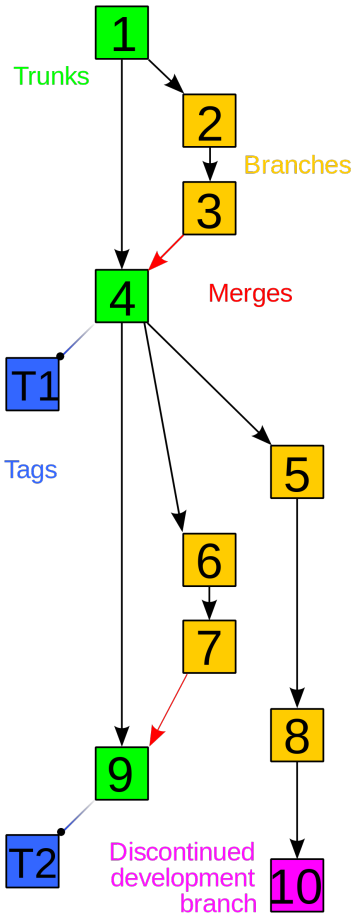


Preface

New to This Edition

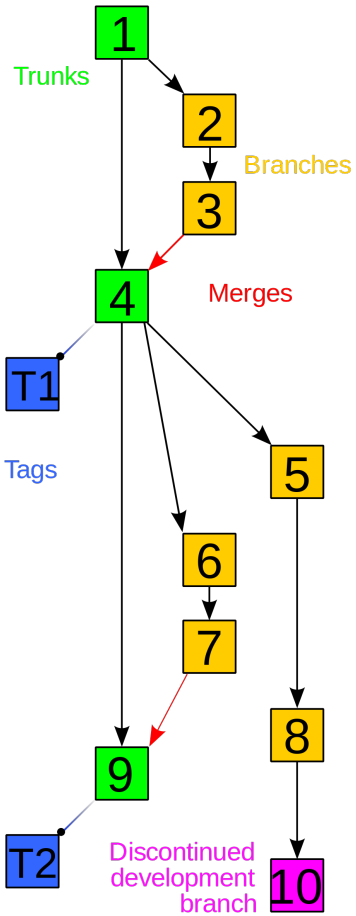
- This eleventh edition of *Modern Labor Economics* has been thoroughly updated in terms of both tabular material and references to the latest literature. Our goal in these updates is to make our textbook a comprehensive reference, for both students and professors, to critical factual information about the labor market *and* to the professional literature in labor economics.
- In recognition of the growing need for rigorous and dispassionate analyses of American immigration policy, we have expanded our analysis of undocumented immigration in chapter 10 to include an enhanced analysis of both its theoretical and measured effects on society.
- We have also incorporated, in relevant chapters, discussions that include labor-market effects of the Great Recession of 2008, along with an examination of recent changes in such outcomes as earnings inequality, human-capital acquisition, and labor-force participation.
- In chapter 6, we added a discussion of the labor supply behavior of married women and a new boxed example on the labor supply of New York City taxi drivers.
- In chapter 11, we amplified the "Group Incentives and Executive Pay" section and added a new boxed example on the "rat race" in law firms.
- In addition to including new material on the recession, we added a new boxed example on earnings inequality in developed countries and a new section on earnings instability to chapter 15.

Principal idea



1 First edition

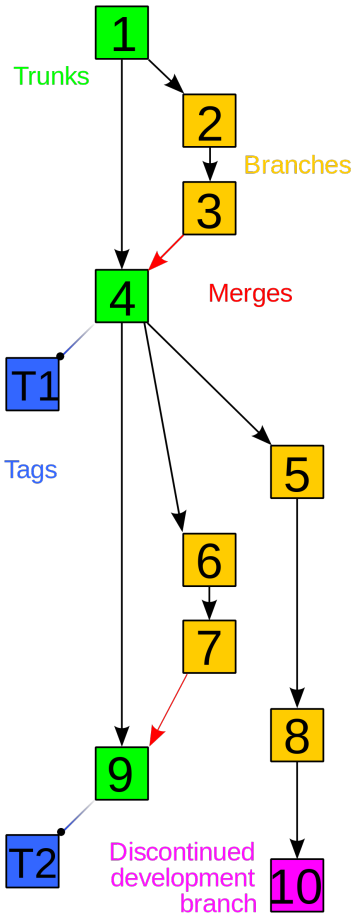
Principal idea



1 First edition

4 Second edition

Principal idea

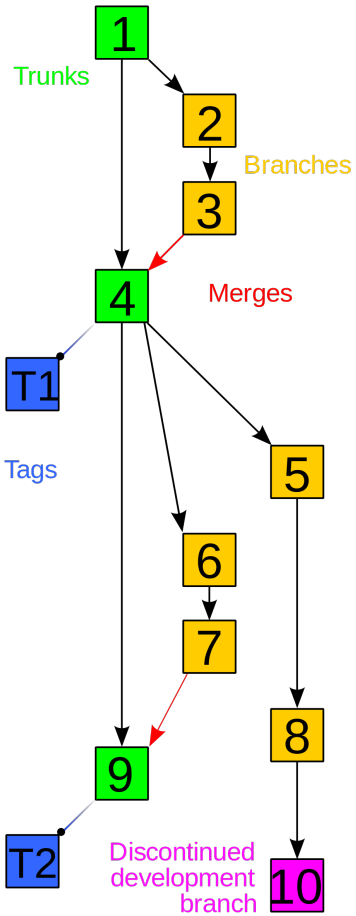


1 First edition

4 Second edition

5 Start of work on a Canadian edition

Principal idea



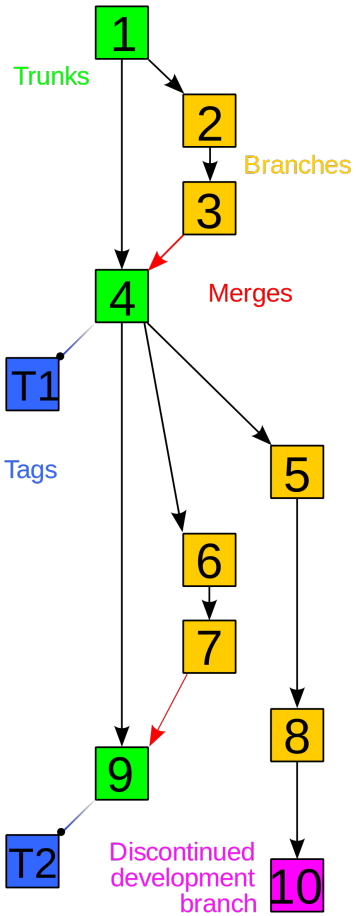
1 First edition

4 Second edition

5 Start of work on a Canadian edition

6 Start of work on the next US edition

Principal idea



1 First edition

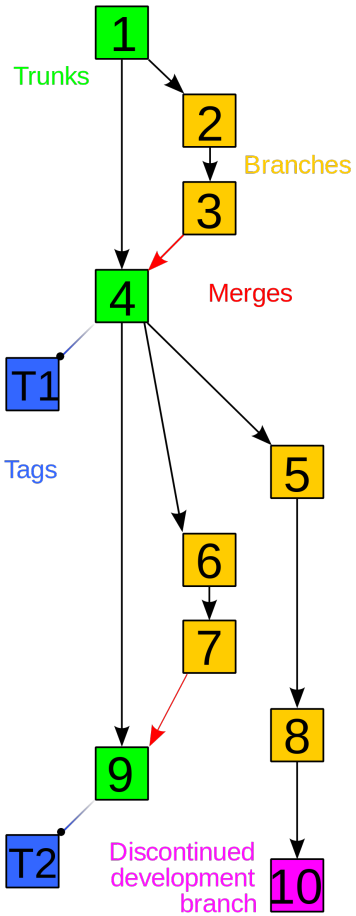
4 Second edition

5 Start of work on a Canadian edition

6 Start of work on the next US edition

9 Third edition

Principal idea



1 First edition

4 Second edition

5 Start of work on a Canadian edition

6 Start of work on the next US edition

9 Third edition

10 First Canadian edition

Two major types of version-management

Centralized model

Server-client approach, editors check out a copy, modify it, and check it back in. Multiple editors:

- *File locking*: only one person can check out any given file
- *Version merging*: discrepancies are handled upon checkin

Two major types of version-management

Centralized model

Server-client approach, editors check out a copy, modify it, and check it back in. Multiple editors:

- *File locking*: only one person can check out any given file
- *Version merging*: discrepancies are handled upon checkin

Distributed model

There is no central server (prescribed by software). Every editor has a full copy of all version, synchronisation occurs by exchanging patches.

Focus in this class

We will focus on one system primarily, and mention a second one:

- Subversion (centralized)

Windows TortoiseSVN (free)
OSX Versions ($), Xcode (free)

- Git (decentralized)

Quickstart on SSG/ECCO

The following really quick tutorial can be run on SSG or on your own computer, if you already have the software installed.

```
ssh NETID@ssg.vrdc.cornell.edu
Last login: Mon Aug 19 15:43:33 2013 from lv39-ws.ilr.cornell.edu
```

```
Welcome to the Social Science Gateway (SSG),
for help, send email to ssg-help@cac.cornell.edu
```

```
ssg:~> svn help
usage: svn <subcommand> [options] [args]
Subversion command-line client, version 1.6.6.
Type 'svn help <subcommand>' for help on a specific subcommand.
Type 'svn --version' to see the program version and RA modules
  or 'svn --version --quiet' to see just the version number.

Most subcommands take file and/or directory arguments, recursing
on the directories.  If no arguments are supplied to such a
command, it recurses on the current directory (inclusive) by default.

Available subcommands:
   add
   blame (praise, annotate, ann)
   cat
[snip]
```

Quickstart Step 1

Let's create a place to work

```
mkdir Workspace
cd Workspace
```

Quickstart Step 1

Let's create a place to work

```
mkdir Workspace
cd Workspace
```

Check out a version of the code:

```
> svn co http://repository.vrdc.cornell.edu/public/test
A    test/test.3
A    test/test.4
A    test/releases
A    test/trunk
A    test/trunk/bls
[snip]
A    test/test.1
Checked out revision 23.
>
```

Checkout To *check out* (or *co*) is to create a local working copy from the repository. A user may specify a specific revision or obtain the latest. The term 'checkout' can also be used as a noun to describe the working copy.

Quickstart 2

Using your favorite editor, modify a file:

```
> cd test
> vi test.1
```

Quickstart 2

Using your favorite editor, modify a file:

```
> cd test
> vi test.1
```

Verify the state of the repsository

```
> svn status
M       test.1
```

Quickstart 2

Using your favorite editor, modify a file:

```
> cd test
> vi test.1
```

Verify the state of the repsository

```
> svn status
M       test.1
```

Let's also add a file:

```
> vi test.today
> svn status
?       test.today
M       test.1
```

The question mark indicates that the file is not (yet) versioned.

Quickstart 3

Add the unversioned file to Subversion tracking

```
> svn add test.today
A         test.today
> svn status
M       test.1
A       test.today
```

Quickstart 3

Add the unversioned file to Subversion tracking

```
> svn add  test.today
A         test.today
> svn status
M       test.1
A       test.today
```

Now we can commit our changes

```
> svn commit
[editor pops up]
Sending        test.1
Adding         test.today
Transmitting file data ..
Committed revision 24.
```

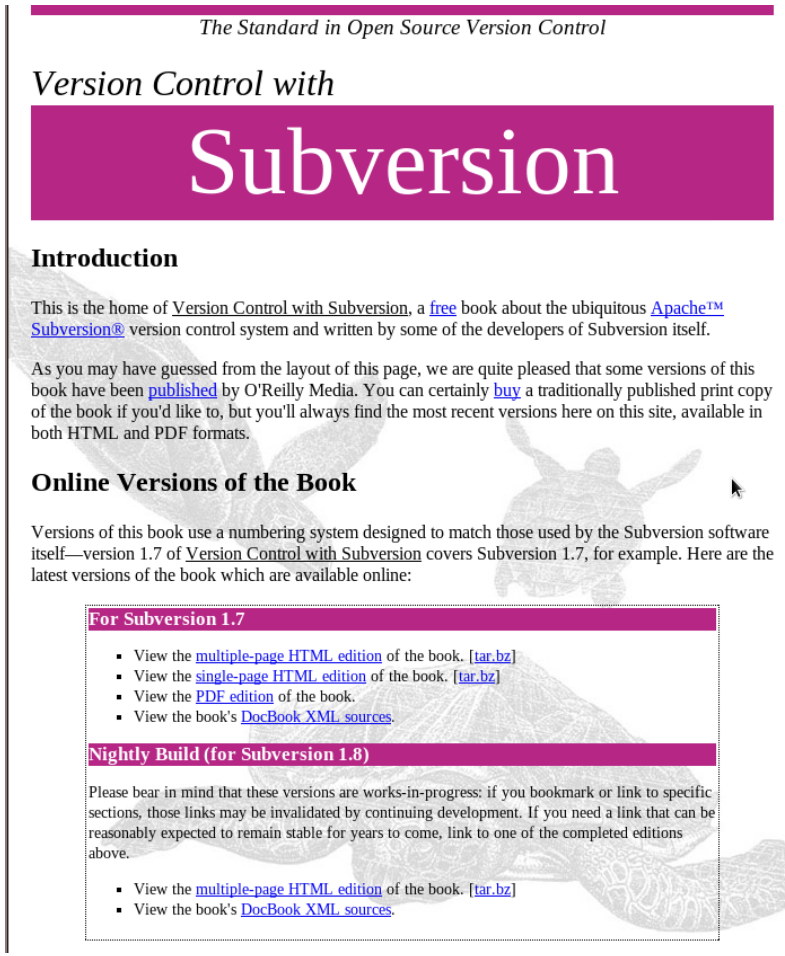
Commit To *commit* (*check in*, *ci*) is to write or merge the changes made in the working copy back to the repository. The terms 'commit' and 'checkin' can also be used as nouns to describe the new revision that is created as a result of committing.

Additional things you will want to learn

Manipulating files

- `svn mv` (vs regular Linux mv or cut-and-paste)
- Reverting changes (`svn revert` before a commit, more complicated after a commit)
- Merging changes (`svn merge`)
- Inspecting logs (`svn log`)
- Identifying changes relative to previous revisions (`svn diff`)

THE reference for Subversion



The Standard in Open Source Version Control

Version Control with

Subversion

Introduction

This is the home of Version Control with Subversion, a free book about the ubiquitous Apache™ Subversion® version control system and written by some of the developers of Subversion itself.

As you may have guessed from the layout of this page, we are quite pleased that some versions of this book have been published by O'Reilly Media. You can certainly buy a traditionally published print copy of the book if you'd like to, but you'll always find the most recent versions here on this site, available in both HTML and PDF formats.

Online Versions of the Book

Versions of this book use a numbering system designed to match those used by the Subversion software itself—version 1.7 of Version Control with Subversion covers Subversion 1.7, for example. Here are the latest versions of the book which are available online:

For Subversion 1.7

- View the multiple-page HTML edition of the book. [tar.bz]
- View the single-page HTML edition of the book. [tar.bz]
- View the PDF edition of the book.
- View the book's DocBook XML sources.

Nightly Build (for Subversion 1.8)

Please bear in mind that these versions are works-in-progress: if you bookmark or link to specific sections, those links may be invalidated by continuing development. If you need a link that can be reasonably expected to remain stable for years to come, link to one of the completed editions above.

- View the multiple-page HTML edition of the book. [tar.bz]
- View the book's DocBook XML sources.

svnbook.red-bean.com

Learning more

A longer tutorial

A longer tutorial (focussing on a complex project setup) is available at the course website.

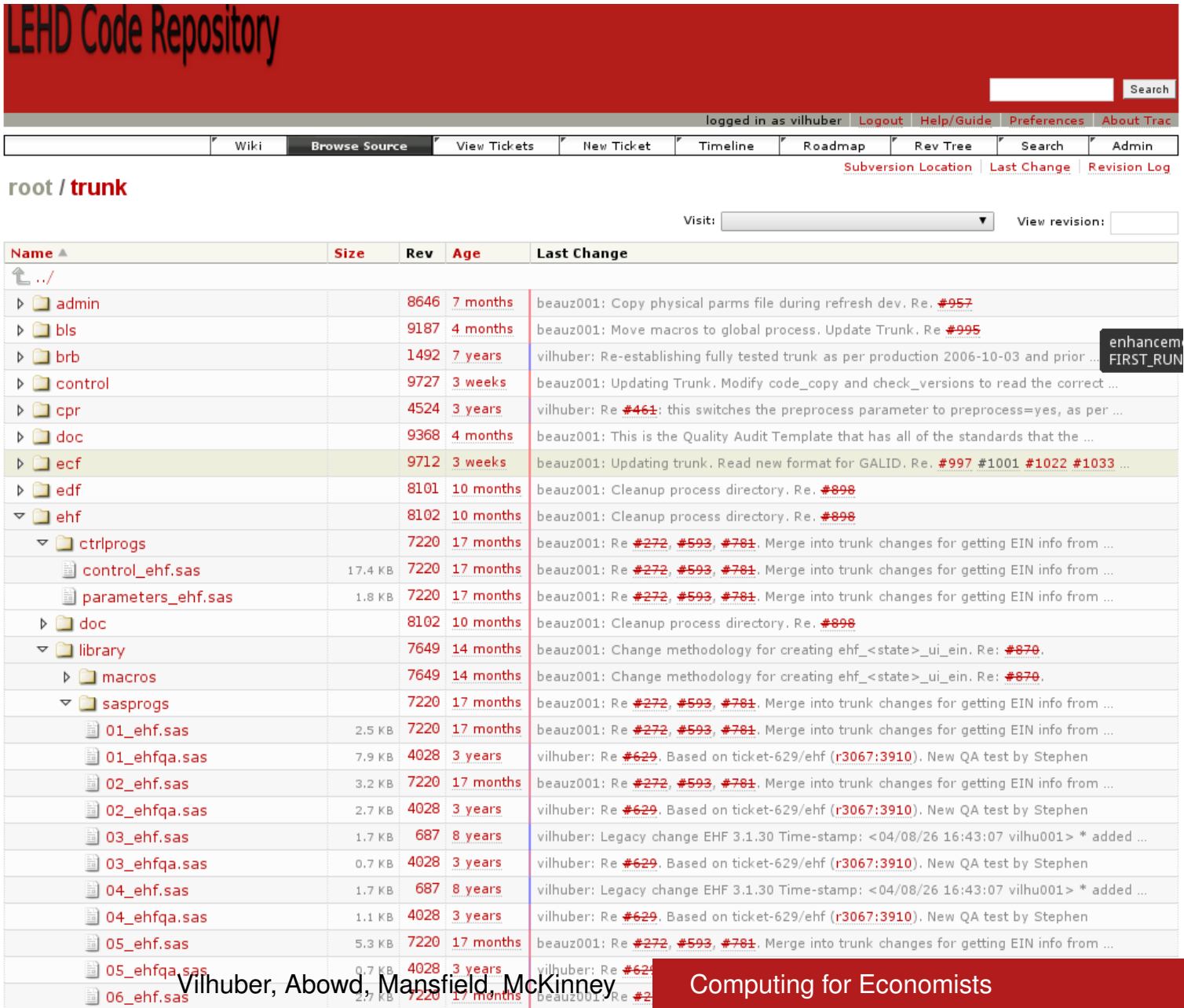
Tracking history

What happened? And who did what?

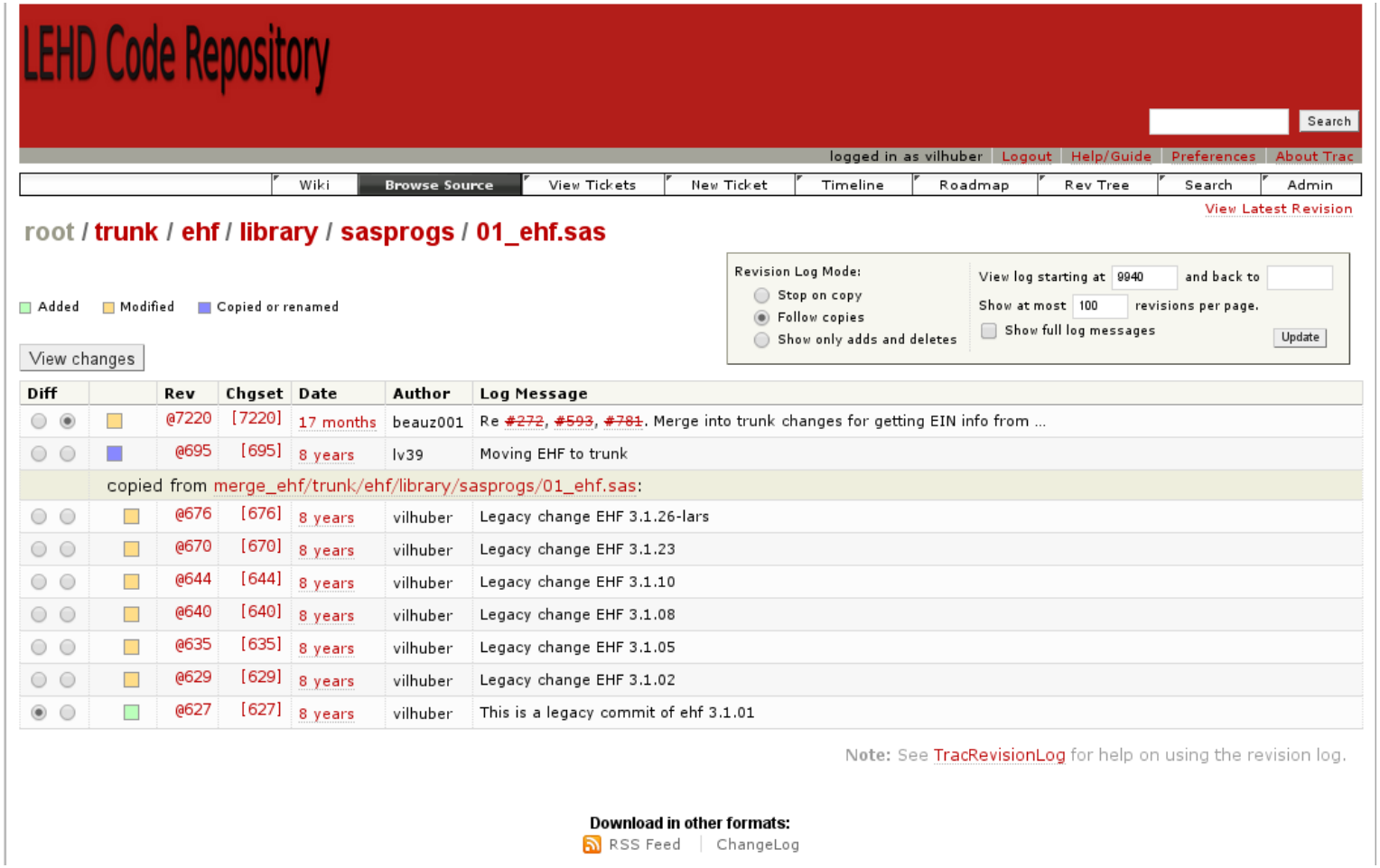
One of the key advantages of using version control systems is ... to control versions.

- Straightforward to view multiple versions of a file (assuming proper usage)
- Possibility to view who changed what ("blame" or "annotate")

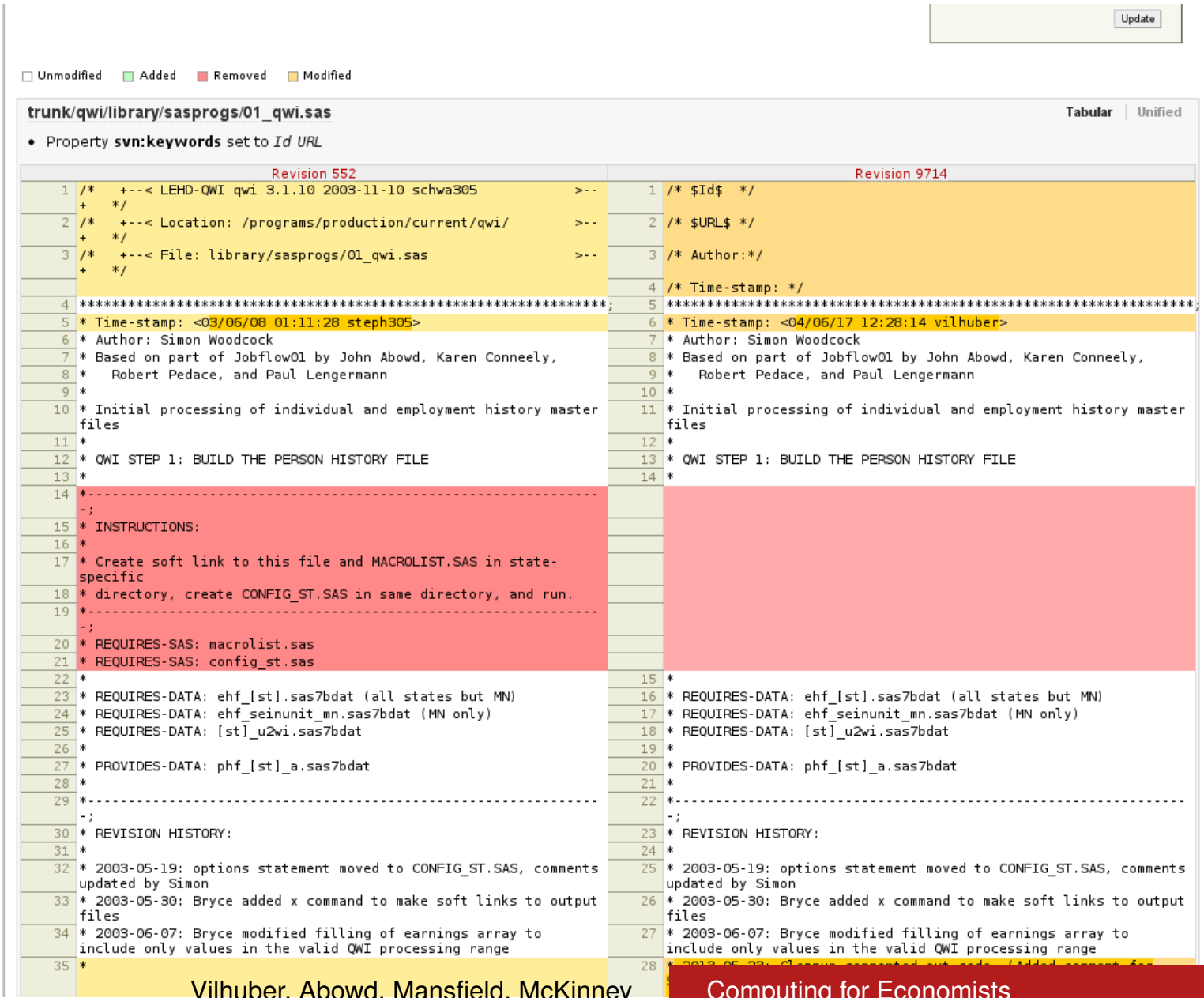
Tracking using web interfaces: SVN



Tracking using web interfaces: SVN



Tracking using web interfaces: SVN



Update

Unmodified Added Removed Modified

trunk/qwi/library/sasprogs/01_qwi.sas Tabular | Unified

- Property **svn:keywords** set to *Id URL*

Revision 552		Revision 9714	
1	/* +--< LEHD-QWI qwi 3.1.10 2003-11-10 schwa305 >-- */	1	/* Id */
2	/* +--< Location: /programs/production/current/qwi/ >-- */	2	/* URL */
3	/* +--< File: library/sasprogs/01_qwi.sas >-- */	3	/* Author:*/
		4	/* Time-stamp: */
4	**;	5	**;
5	* Time-stamp: <03/06/08 01:11:28 steph305>	6	* Time-stamp: <04/06/17 12:28:14 vilhuber>
6	* Author: Simon Woodcock	7	* Author: Simon Woodcock
7	* Based on part of Jobflow01 by John Abowd, Karen Conneely,	8	* Based on part of Jobflow01 by John Abowd, Karen Conneely,
8	* Robert Pedace, and Paul Lengermann	9	* Robert Pedace, and Paul Lengermann
9	*	10	*
10	* Initial processing of individual and employment history master files	11	* Initial processing of individual and employment history master files
11	*	12	*
12	* QWI STEP 1: BUILD THE PERSON HISTORY FILE	13	* QWI STEP 1: BUILD THE PERSON HISTORY FILE
13	*	14	*
14	*--;		
15	* INSTRUCTIONS:		
16	*		
17	* Create soft link to this file and MACROLIST.SAS in state-specific		
18	* directory, create CONFIG_ST.SAS in same directory, and run.		
19	*--;		
20	* REQUIRES-SAS: macrolist.sas		
21	* REQUIRES-SAS: config_st.sas		
22	*	15	*
23	* REQUIRES-DATA: ehf_[st].sas7bdat (all states but MN)	16	* REQUIRES-DATA: ehf_[st].sas7bdat (all states but MN)
24	* REQUIRES-DATA: ehf_seinunit_mn.sas7bdat (MN only)	17	* REQUIRES-DATA: ehf_seinunit_mn.sas7bdat (MN only)
25	* REQUIRES-DATA: [st]_u2wi.sas7bdat	18	* REQUIRES-DATA: [st]_u2wi.sas7bdat
26	*	19	*
27	* PROVIDES-DATA: phf_[st]_a.sas7bdat	20	* PROVIDES-DATA: phf_[st]_a.sas7bdat
28	*	21	*
29	*--;	22	*--;
30	* REVISION HISTORY:	23	* REVISION HISTORY:
31	*	24	*
32	* 2003-05-19: options statement moved to CONFIG_ST.SAS, comments updated by Simon	25	* 2003-05-19: options statement moved to CONFIG_ST.SAS, comments updated by Simon
33	* 2003-05-30: Bryce added x command to make soft links to output files	26	* 2003-05-30: Bryce added x command to make soft links to output files
34	* 2003-06-07: Bryce modified filling of earnings array to include only values in the valid QWI processing range	27	* 2003-06-07: Bryce modified filling of earnings array to include only values in the valid QWI processing range
35	*	28	* 2013-05-23: Cleanup, commented out code. (Added comment for

Tracking using web interfaces: SVN

LEHD Code Repository

Search

logged in as vilhuber | Logout | Help/Guide | Preferences | About Trac

Wiki | Browse Source | View Tickets | New Ticket | Timeline | Roadmap | Rev Tree | Search | Admin

Subversion Location | Last Change | Normal | Revision Log

root / trunk / qwi / library / sasprogs / 01_qwi.sas @ 9714

Visit: | View revision: 9714

Revision 9714, 1.8 KB (checked in by beauz001, 3 weeks ago)

Updating Trunk. Re: #433 #554 #1003 #1010 #1016 #1020 #1021 #1026 #1031 #1032 #1044 #1054 #1056

Property **svn:executable** set to *

Property **svn:keywords** set to *Id URL*

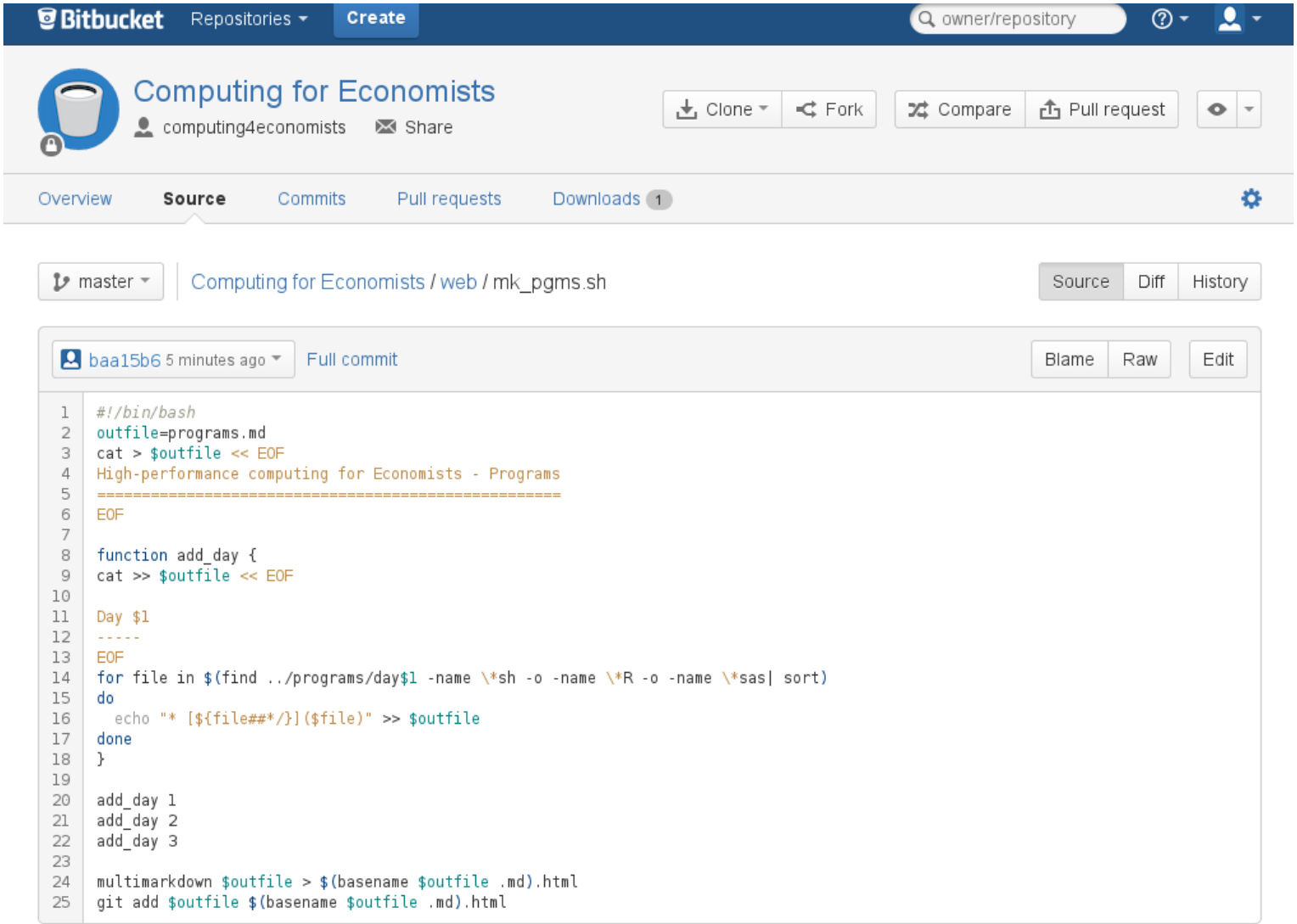
Rev	Line	
[7204]	1	/* Id */
[6733]	2	/* URL */
	3	/* Author:*/
	4	/* Time-stamp: */
[558]	5	**;
[576]	6	* Time-stamp: <04/06/17 12:28:14 vilhuber>
[558]	7	* Author: Simon Woodcock
	8	* Based on part of Jobflow01 by John Abowd, Karen Conneely,
	9	* Robert Pedace, and Paul Lengermann
	10	*
	11	* Initial processing of individual and employment history master files
	12	*
	13	* QWI STEP 1: BUILD THE PERSON HISTORY FILE
	14	*
	15	*
	16	* REQUIRES-DATA: ehf_[st].sas7bdat (all states but MN)
	17	* REQUIRES-DATA: ehf_seinunit_mn.sas7bdat (MN only)
	18	* REQUIRES-DATA: [st]_u2wi.sas7bdat
	19	*
	20	* PROVIDES-DATA: phf_[st]_a.sas7bdat
	21	*
	22	*--;
	23	* REVISION HISTORY:
	24	*
	25	* 2003-05-19: options statement moved to CONFIG_ST.SAS, comments updated by Simon
	26	* 2003-05-30: Bryce added x command to make soft links to output files
	27	* 2003-06-07: Bryce modified filling of earnings array to include only values in the valid QWI processing range
[9714]	28	* 2013-05-23: Cleanup commented out code. (Added comment for Stephen Tibbets change.)
[558]	29	**;
	30	
	31	
	32	title1 "Create the Person History File (PHF)";
	33	title2 "/programs/projects/EDE/v3.0/&st./01.create_phf.sas";
	34	
	35	

Tracking using web interfaces: SVN

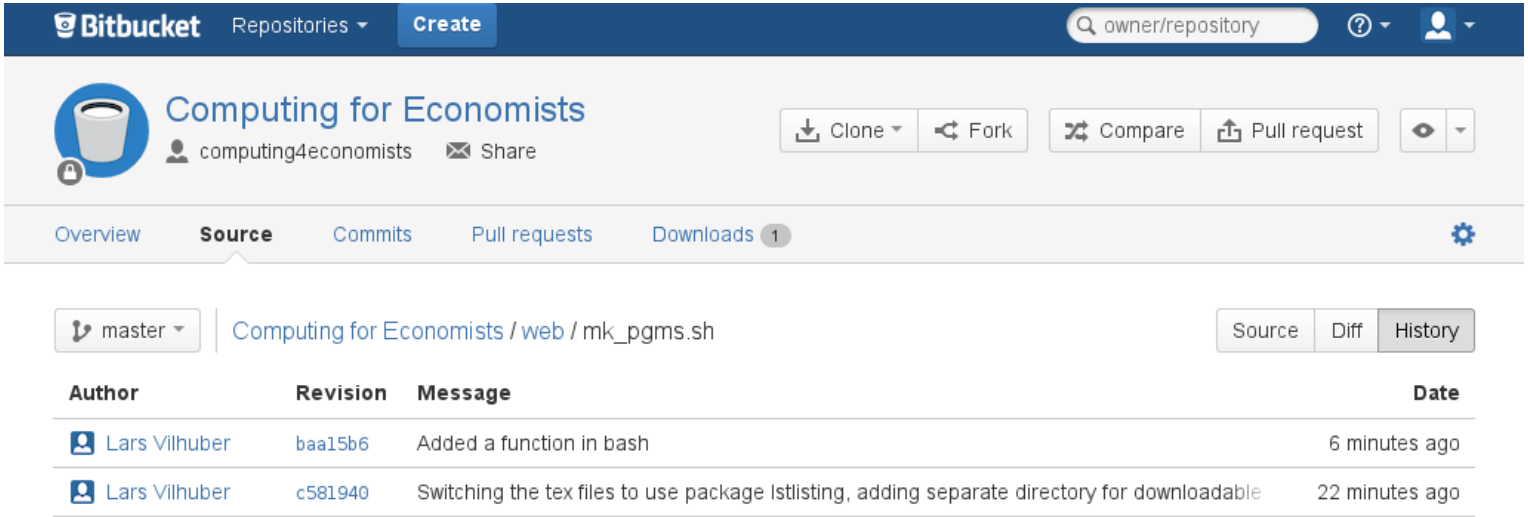
Including our demo change

VRDC repository viewer

Tracking using web interfaces: git



Tracking using web interfaces: git



Tracking using web interfaces: git

master | Computing for Economists / web / mk_pgms.sh Source | Diff | History

Diff from c581940 22 minutes ago to baa15b6 6 minutes ago

web/mk_pgms.sh Side-by-side diff

```
...
3  3   cat > $outfile << EOF
4  4   High-performance computing for Economists - Programs
5  5   ==================================================
   6  +EOF
   7  +
   8  +function add_day {
   9  +cat >> $outfile << EOF
6  10
7     -Day 1
   11 +Day $1
8  12  -----
9  13  EOF
10    -
11    -for file in $(find ../programs/day1 -name \*sh -o -name \*R -o -name \*sas| sort)
   14 +for file in $(find ../programs/day$1 -name \*sh -o -name \*R -o -name \*sas| sort)
12 15  do
13 16    echo "* [${file##*/}]($file)" >> $outfile
14 17  done
   18 +}
   19 +
   20 +add_day 1
   21 +add_day 2
   22 +add_day 3
15 23
16 24  multimarkdown $outfile > $(basename $outfile .md).html
17 25  git add $outfile $(basename $outfile .md).html
...
```

Subversion vs. Git

incomplete

Checking out a subdirectory

SVN

```
svn co ${URL}/sub/directory (name)
```

GIT

```
mkdir <repo> && cd <repo>
git init
git remote add -f <name> <url>
git config core.sparsecheckout true
echo some/dir/ >> .git/info/sparse-checkout
echo another/sub/tree >>
.git/info/sparse-checkout
git pull <remote> <branch>
```

Source: here

VCS infrastructure

At Cornell

Cornell Sourceforge (see instructions on how to create)

Elsewhere

- CloudForge (free for personal use)
- GitHub (Git, subversion, free for open-source)
- BitBucket (Git, no subversion, free for academic users)
- Roll your own: very simple webserver setup for Subversion, no server required for Git on your PC

Next section