

We're building the GNU system!

Ludovic Courtès

`ludo@gnu.org`

GNU Hackers Meeting  
15 August 2014, München

Howdy!



Howdy!



Howdy!



the “**GNU system**”,  
aka. “a GNU distribution”

# the “GNU system”, aka. “a GNU distribution”

- ▶ protect & enhance **computing freedom**
- ▶ improve **integration** of GNU software, consistency
- ▶ improve **workflow** among GNU hackers & users

Dependable. Hackable. Liberating.

**Dependable.**



per-user, transactional package installation etc.

```
alice@foo$ guix package --install=gcc  
alice@foo$ guix gc --references 'which gcc'  
/gnu/store/...-glibc-2.19  
/gnu/store/...-gcc-4.9.1  
...
```

**demo!**

```
bob@foo$ guix package --install=gcc-4.7.3  
bob@foo$ guix gc --references 'which gcc'  
/gnu/store/...-glibc-2.17  
/gnu/store/...-gcc-4.7.3  
...
```

# transparent binary/source deployment

```
alice@foo$ guix package --install=emacs
```

```
The following package will be installed:
```

```
  emacs-24.3 out /gnu/store/...-emacs-24.3
```

```
The following files will be downloaded:
```

```
  /gnu/store/...-emacs-24.3
```

```
  /gnu/store/...-libxpm-3.5.10
```

```
  /gnu/store/...-libxext-1.3.1
```

```
  /gnu/store/...-libxaw-1.0.11
```

# transparent binary/source deployment

```
alice@foo$ guix package --install=emacs
```

```
The following package will be installed:
```

```
  emacs-24.3 out /gnu/store/...-emacs-24.3
```

```
The following files will be downloaded:
```

```
  /gnu/store/...-libxext-1.3.1
```

```
  /gnu/store/...-libxaw-1.0.11
```

```
The following derivations will be built:
```

```
  /gnu/store/...-emacs-24.3.drv
```

```
  /gnu/store/...-libxpm-3.5.10.drv
```

workflow

```
(define foo (package ...))
```

**user**

# workflow

```
(define foo (package ...))
```

test

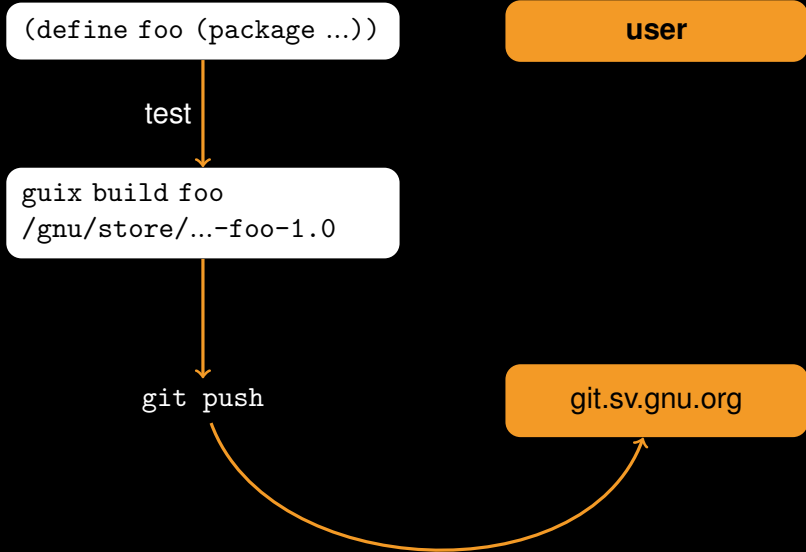


```
guix build foo  
/gnu/store/...-foo-1.0
```

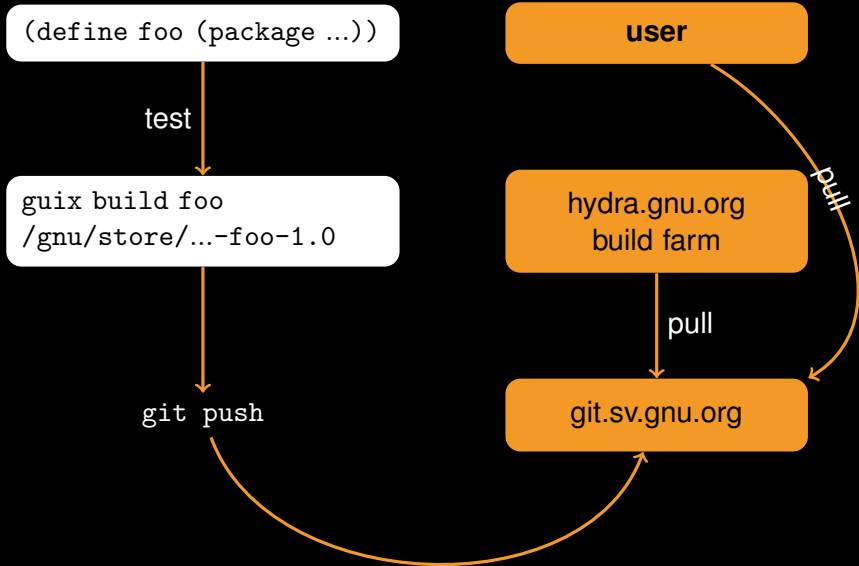
**user**



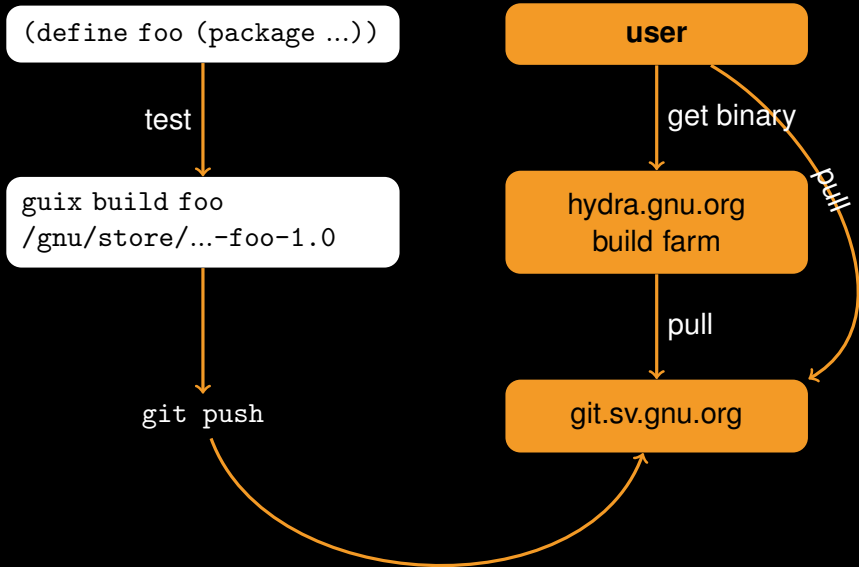
# workflow



# workflow

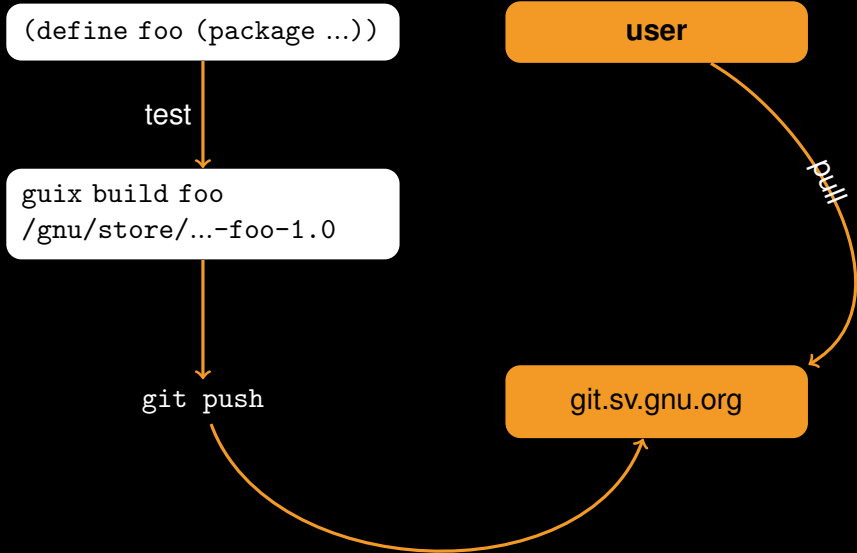


# workflow

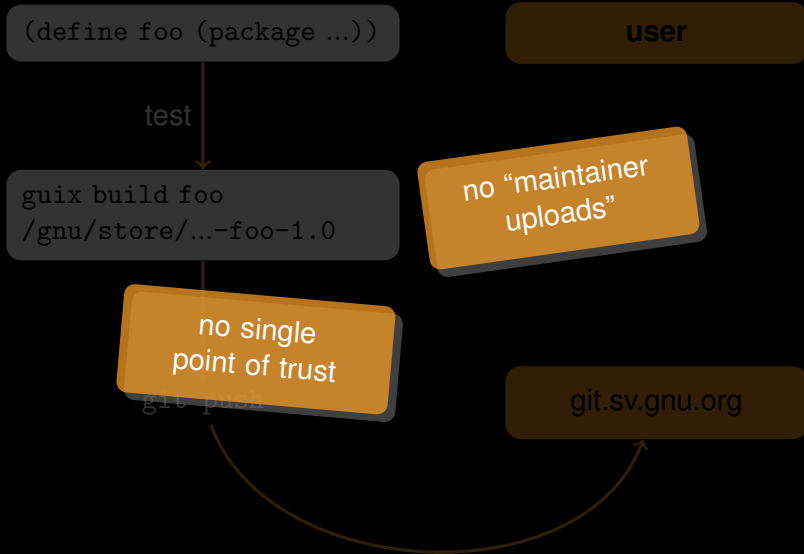




# workflow



# workflow



does this binary **correspond**  
to that source?

towards deterministic builds


```
$ guix build guile
```

**isolated build:** chroot, separate name spaces, etc.

## towards deterministic builds

```
$ guix build guile  
/gnu/store/ h2g4sc09h4... -guile-2.0.9
```

hash of **all** the dependencies



## towards deterministic builds

```
$ guix build guile
/gnu/store/h2g4sc09h4...-guile-2.0.9

$ guix gc --references /gnu/store/...-guile-2.0.9
/gnu/store/4jl83jgzaac...-glibc-2.17
/gnu/store/iplay43cg58...-libunistring-0.9.3
/gnu/store/47p47v92cj9...-libffi-3.0.9
/gnu/store/drkwck2j965...-gmp-5.0.5
...
```

## towards deterministic builds

```
$ guix build guile  
/gnu/store/h2g4sc09h4...-guile-2.0.9
```

```
$ guix gc --references /gnu/store/...-guile-2.0.9  
/gnu/store/4jl83jgzaac...-glibc-2.17  
/gnu/store/iplay43cg58...-libunistring-0.9.3  
/gnu/store/47p47v92cj9...-libffi-3.0.9  
/gnu/store/drkwck2j9...-gmp-6.0.0  
...
```

**(nearly) bit-identical for everyone**



## The Hacking Process

1. (R)econnaissance
2. (I)nfection
3. (C)ommand And Control
4. (E)xfiltration





## The Hacking Process

1. (R)econnaissance

2. do not **trust** a single  
3. binary provider

4. (E)xfiltration

# Deterministic Builds: Integrity through Decentralization

– Mike Perry

cool features for  
**GNU maintainers**

```
$ guix package -i gettext
```

```
looking for latest release of GNU gettext...
```

```
$ guix package -i gettext
```

```
note: using 0.18.3 but 0.19.2 is available upstream
```

```
$ guix build emacs
```

```
$ guix build emacs \  
--with-source=ftp://alpha.gnu....emacs-24.3.92.tar.xz
```

```
$ guix build emacs \  
--with-source=ftp://alpha.gnu.../emacs-24.3.92.tar.xz
```

**pre-release testing in  
a pristine environment**



From: **Daiki Ueno** <ueno@gnu.org>  
Subject: libunistring-0.9.4 release candidate  
Newsgroups: *gmane.comp.gnu.libunistring.bugs*  
To: *bug-libunistring@gnu.org*  
Date: *Tue, 12 Aug 2014 16:36:06 +0900 (2 days, 7 hours, 44 minutes ago)*  
User-Agent: *Gnus/5.13 (Gnus v5.13) Emacs/24.4.50 (gnu/linux)*  
Xref: *news.gmane.org gmane.comp.gnu.libunistring.bugs:114*

Hello,

As some of you might know, I'll be taking care of the library for now (co-maintainers are welcome). Before applying big changes such as updating to the latest Unicode standard, I thought it would be a good idea to make a new release from the current git master.

So, I've generated a pre-release tarball:  
<http://alpha.gnu.org/gnu/libunistring/libunistring-0.9.4-rc1.tar.gz>

I applied a couple of minor build fixes, but if you find anything else which should be included in the release, please report to the list.

Regards,  
--  
*Daiki Ueno*

From: **Daiki Ueno** <ueno@gnu.org>  
Subject: libunistring-0.9.4 release candidate  
Newsgroups: *gmane.comp.gnu.libunistring.bugs*  
To: **bug-libunistring@gnu.org**  
Date: *Tue, 12 Aug 2014 16:36:06 +0900 (2 days, 7 hours, 44 minutes ago)*  
User-Agent: *Gnus/5.13 (Gnus v5.13) Emacs/24.4.50 (gnu/linux)*  
Xref: *news.gmane.org gmane.comp.gnu.libunistring.bugs:114*

Hello,

As some of you might know, I'll be taking care of the library for now (co-maintainers are welcome). Before applying changes such as updating to the latest libc version, I thought it would be a good idea to make a new release from the current git master.

**send a package recipe! :-)**

So, I've generated a pre-release tarball:

<http://alpha.gnu.org/gnu/libunistring/libunistring-0.9.4-rc1.tar.gz>

I applied a couple of minor build fixes, but if you find anything else which should be included in the release, please report to the list.

Regards,

--

*Daiki Ueno*

```
(use-modules (guix) (gnu))
```

```
(package (inherit
```

```
          (car (find-packages-by-name "libunistring")))
```

```
  (version "0.9.4rc1")
```

```
  (source (origin
```

```
            (method url-fetch)
```

```
            (uri "http://.../libunistring-0.9.4-rc1.tar.gz")
```

```
            (sha256 (base32 "14pi90...")))))
```

```
(use-modules (guix) (gnu))
```

```
(package (inherit  
          (car (find-packages-by-name "libunistring")))  
  (version "0.9.4rc1")  
  (source (origin  
            (method url-fetch)  
            (uri "http://.../libunistring-0.9.4-rc1.tar.gz")  
            (sha256 (base32 "14pi90...")))))
```

```
$ guix build -e '(load "libunistring-rc.scm")'
```

The following derivations will be built:

```
/gnu/store/...-libunistring-0.9.4rc1.drv
```

```
/gnu/store/...-libunistring-0.9.4-rc1.tar.gz.drv
```

```
(use-modules (guix) (gnu))
```

```
(package (inherit  
          (car (find-packages-by-name "libunistring")))  
  (version "0.9.4rc1")  
  (source (origin  
            (method url-fetch)  
            (uri "http://.../libunistring-0.9.4-rc1.tar.gz")  
            (sha256 (base32 "14p190..."))))
```

**but does, say, Guile work with the RC?**

```
$ guix build -e '(load "libunistring-rc.scm")'
```

The following derivations will be built:

```
/gnu/store/...-libunistring-0.9.4rc1.drv
```

```
/gnu/store/...-libunistring-0.9.4-rc1.tar.gz.drv
```

```
(use-modules (gnu packages guile) (guix)
             (srfi srfi-1))
```

```
(package (inherit guile-2.0)
```

```
;; Replace the stable libunistring with the RC
```

```
;; in Guile's dependencies.
```

```
(inputs '(("libunistring"
          ,(primitive-load "libunistring-rc.scm"))
        ,@(alist-delete "libunistring"
                       (package-inputs guile-2.0))))))
```

```
(use-modules (gnu packages guile) (guix)
             (srfi srfi-1))
```

```
(package (inherit guile-2.0)
  ;; Replace the stable libunistring with the RC
  ;; in Guile's dependencies.
  (inputs '(("libunistring"
             ,(primitive-load "libunistring-rc.scm"))
            ,@(alist-delete "libunistring"
                            (package-inputs guile-2.0))))))
```

```
$ guix build -e '(load "guile-libunistring.scm")'
```

The following derivations will be built:

```
/gnu/store/...-guile-2.0.11.drv
```

```
/gnu/store/...-libunistring-0.9.4-rc1.tar.gz.drv
```

```
/gnu/store/...-libunistring-0.9.4rc1.drv
```

```
(use-modules (gnu packages guile) (guix)
             (srfi srfi-1))
```

```
(package (inherit guile-2.0)
```

```
;; Replace the stable libunistring with the RC
```

```
;; in Guile's dependencies.
```

```
(inputs '(("libunistring"
```

```
          ,(primitive-load "libunistring-rc.scm"))
```

```
          ,@(alist-delete "libunistring"
```

```
                        (package-inputs guile-2.0))))
```

**reliable integration testing**

```
$ guix build -e '(load "guile-libunistring.scm")'
```

The following derivations will be built:

```
/gnu/store/...-guile-2.0.11.drv
```

```
/gnu/store/...-libunistring-0.9.4-rc1.tar.gz.drv
```

```
/gnu/store/...-libunistring-0.9.4rc1.drv
```



# even better

1. **iterate** over the packages with  
fold-packages
2. identify those **depending on libunistring**
3. compute **package variants** that use the RC
4. **build them**

# even better

(left as an exercise to the audience)

1. **iterate** over the packages with  
fold-packages
2. identify those **depending on libunistring**
3. compute **package variants** that use the RC
4. **build them**

**Hackable.**

The truth is that Lisp is not the right language for any particular problem. Rather, Lisp encourages one to attack a new problem by **implementing new languages** tailored to that problem.

– Abelson & Sussman, 1987

```
(define hello
  (package
    (name "hello")
    (version "2.8")
    (source (origin
              (method url-fetch)
              (uri (string-append
                    "mirror://gnu/.../hello-" version
                    ".tar.gz"))
              (sha256 (base32 "0wqd...dz6")))))
  (build-system gnu-build-system)
  (synopsis "Hello, GNU world: An example GNU package")
  (description "Produce a friendly greeting.")
  (home-page "http://www.gnu.org/software/hello/")
  (license gpl3+)))
```

**build processes**  
chroot, separate UIDs

**Guile**

(guix packages)

(guix store)

**build daemon**

**build processes**  
chroot, separate UIDs

**Guile**

(guix packages)

(guix store)

**build daemon**

RPCs

```
graph TD; Guile["Guile  
(guix packages)  
(guix store)"] -- RPCs --> BuildDaemon["build daemon"]; subgraph BuildProcesses ["build processes"]; direction TB; subgraph Env ["chroot, separate UIDs"]; direction TB; B["build processes"]; end; end;
```

**build processes**  
chroot, separate UIDs

Guile, make, etc.

Guile, make, etc.

Guile, make, etc.

**Guile**

(guix packages)

(guix store)

**build daemon**

RPCs

```
graph TD; subgraph BuildProcesses [build processes]; direction TB; G1[Guile, make, etc.]; G2[Guile, make, etc.]; G3[Guile, make, etc.]; end; Guile[Guile  
(guix packages)  
(guix store)]; BuildDaemon[build daemon]; Guile -- RPCs --> BuildDaemon; BuildDaemon --> BuildProcesses;
```



```
(use-modules (guix) (gnu packages base))
```

```
,enter-store-monad
```

```
(interned-file "README")
```

```
=> "/gnu/store/rwmzi9j1j77a7bq5kyiy2abdqznr3f02-README"
```

```
(gexp->derivation "list-files"
```

```
  #~(symlink (string-append #scoreutils "/bin/ls")
```

```
    #output
```

**Emacs +  
Geiser demo!**

```
=> #<derivation "/gnu/store/xyz...-list-files.drv" ...>
```

```
(package->derivation hello)
```

```
=> #<derivation "/gnu/store/xyz...-hello-2.8.drv" ...>
```

```
(built-derivations (list drv))
```

```
... daemon builds/downloads package on our behalf...
```

```
=> #t
```

copy fields from hello except  
for version and source

```
(package ( inherit hello)
  (version "2.7")
  (source
    (origin
      (method url-fetch)
      (uri "mirror://gnu/hello/hello-2.7.tar.gz")
      (sha256
        (base32 "7dqw3..."))))))
```

```
(define (static-package p)
  ;; Return a statically-linked variant of P.
  (package (inherit p)
    (arguments
      '(:configure-flags '("--disable-shared"
                            "LDFLAGS=-static")
        ,@(package-arguments p))))))
```

**and now, the operating system**

```
(define my-config
  (operating-system
    (host-name "gnubox")
    (timezone "Europe/Paris")
    (locale "en_US.UTF-8")
    (bootloader (grub-configuration (device "/dev/sdX"))))

  (file-systems (cons (file-system
                       (mount-point "/") ...)
                      %base-file-systems))

  (users (list (user-account
                (name "ludo") (group "users")
                (comment "Hello, this is me!")
                (home-directory "/home/ludo"))))

  (packages (cons* iotop jnettop %base-packages)))
```

```
(define my-config
  (operating-system
    (host-name "gnubox")
    (timezone "Europe/Paris")
    (locale "en_US.UTF-8")
    (bootloader (grub-configuration (device "/dev/sdX")))
    (initrd (cut base-initrd <>
              #:extra-modules '("usb-storage.ko")))
    (file-systems (cons (file-system
                        (mount-point "/" ...)
                        %base-file-systems))
                    (users (list (user-account
                                  (name "ludo") (group "users")
                                  (comment "Hello, this is me!")
                                  (home-directory "/home/ludo"))))
                    (packages (cons* iotop jnettop %base-packages)))
```

```
$ guix system build os-config.scm  
/gnu/store/lq5kbp84kl9q9ncbk578wa7x5x0lr13f-system
```

```
$ guix system vm os-config  
/gnu/store/yan79p9k83dww5yd6jgmhph0nxalfn6-run-vm.sh
```

```
# guix system init os-config /mnt  
...
```

```
# guix system reconfigure os-config  
...
```



**demo!**

PID 1 is  
GNU dmd!

```
# deco status dmd
```

```
Started: (term-tty1 term-tty2 nscd syslog)
```

```
Stopped: ()
```

```
# deco stop nscd
```

```
Service nscd has been stopped
```



# GNU dmd in a nutshell

- ▶ born in 2003, revived in 2013 :-)
- ▶ dependency-based service manager

# GNU dmd in a nutshell

- ▶ born in 2003, revived in 2013 :-)
- ▶ dependency-based service manager
- ▶ dmd is PID 1, deco is a client

# GNU dmd in a nutshell

- ▶ born in 2003, revived in 2013 :-)
- ▶ dependency-based service manager
- ▶ dmd is PID 1, deco is a client
- ▶ written in Guile Scheme
- ▶ dynamic, extensible, etc.

extra coolness



# GNU/Hurd port—the true GNU!

WIP by Manolis Ragkousis

- ▶ step #1: cross-compiling to GNU/Hurd

# GNU/Hurd port—the true GNU!

WIP by Manolis Ragkousis

- ▶ step #1: cross-compiling to GNU/Hurd
- ▶ packaged Mach, MiG, the Hurd, and...
- ▶ ... libc

# GNU/Hurd port—the true GNU!

WIP by Manolis Ragkousis

- ▶ step #1: cross-compiling to GNU/Hurd
- ▶ packaged Mach, MiG, the Hurd, and...
- ▶ ... libc
- ▶ but! GNU libc **fails to build** on GNU/Hurd

# GNU/Hurd port—the true GNU!

WIP by Manolis Ragkousis

- ▶ step #1: cross-compiling to GNU/Hurd
- ▶ packaged Mach, MiG, the Hurd, and...
- ▶ ... libc
- ▶ but! GNU libc **fails to build** on GNU/Hurd
- ▶ (Debian's version does build...)



# GNU/Hurd port—the true GNU!

WIP by Manolis Ragkousis

- ▶ step #1: cross-compiling to GNU/Hurd
- ▶ packaged Mach, MiG, the Hurd, and...
- ▶ ... libc **help the Hurd & libc**
- ▶ but! GNU **regain harmony!**
- ▶ (Debian's version does build...)



## Packages 12

Name	Version	Synopsis	Home Page	License
<a href="#">bigloo</a>	4.1a	Bigloo, an efficient Scheme compiler	<a href="http://www-sop.inria.fr/index/fp/Bigloo/">http://www-sop.inria.fr/index/fp/Bigloo/</a>	GPL 2+
<a href="#">chicken</a>	4.8.0.3	R5RS Scheme implementation that compiles native code via C	<a href="http://www.call-cc.org/">http://www.call-cc.org/</a>	Modified BSD
<a href="#">guile</a>	1.8.8	Scheme implementation intended especially for extensions	<a href="http://www.gnu.org/software/guile/">http://www.gnu.org/software/guile/</a>	LGPL 2.0+
<a href="#">guile</a>	2.0.9	Scheme implementation intended especially for	<a href="http://www.gnu.org/software/guile/">http://www.gnu.org/software/guile/</a>	LGPL 3+

**World  
première!**



WIP by David Thompson

<https://gitorious.org/guix-web/guix-web>

## Packages 12

Name	Version	Synopsis	Home Page	License
<a href="#">bigloo</a>	4.1a	Bigloo, an efficient Scheme compiler	<a href="http://www-sop.inria.fr/index/fp/Bigloo/">http://www-sop.inria.fr/index/fp/Bigloo/</a>	GPL 2+
<a href="#">chicken</a>	4.8.0.3	R5RS Scheme implementation that compiles native code via C	<a href="http://www.call-cc.org/">http://www.call-cc.org/</a>	Modified BSD
<a href="#">guile</a>	1.8.8	Scheme implementation intended especially for extensions	<a href="http://www.gnu.org/software/guile/">http://www.gnu.org/software/guile/</a>	LGPL 2.0+
<a href="#">guile</a>	2.0.9	Scheme implementation intended especially for	<a href="http://www.gnu.org/software/guile/">http://www.gnu.org/software/guile/</a>	LGPL 3+

Name ▼	Version	Outputs	Installed
bigloo	4.1a	out	—
chicken	4.8.0.3	out	—
geiser	0.6	ou	—
guile	1.8.8	out	—
guile	2.0.11	out, debug	out
guile	2.0.11	out, debug	out
guile-cairo	1.4.1	out	—
guile-lib	0.2.2	out	—
guile-reader-for-...	0.6	out	—
guile-reader-for-...	0.6	out	—
guile-static-stri...	2.0.11	out	—
guile-static-stri...	2.0.11	out	—
guile-wm	1.0	out	out
guile-xcb	1.3	out	—
mcron	1.0.8	out	—
mit-scheme	9.1.1	out	out
racket	5.3.4	out	—
scheme48	1.9	out	—
skribilo	0.9.2	out	—

A yellow banner with a black border and a slight shadow, containing the text "guix.el!" in a bold, black, sans-serif font. The banner is tilted slightly upwards to the right and is positioned over the middle of the table.

Name : guile  
Version : 2.0.11  
Synopsis : Scheme implementation intended especially  
for extensions



Outputs:  
out   
Installed path:  
[/gnu/store/h8d3hinaiwmvxbhk9g9b67s0bw7v8d3r-guile-2.0.](#)

11  
Dependencies:  
[/gnu/store/bda98x7qxsd90qz4kr4rd03gbqsd038x-libunistring-0.9.3](#)  
[/gnu/store/j2ainmhcqcb5hgcqanywdzci84dns5lf-libtool-2.](#)  
4.2  
[/gnu/store/lfgzbha6s711hynxdcj0xcd4l17582sy-libgc-7.4.](#)  
0  
[/gnu/store/84xqp874r7a65alwvl11g6znvk90812n-gmp-6.0.0a](#)

debug

```
Name      : guile
Version   : 2.0.11
Synopsis  : Scheme implementation intended especially
          for extensions
Outputs:
  out      Delete https://github.com/alezost/guix.el
           Installed path: (merged soon)
           /gnu/store/h8d3hinaiwmvxbhk9g9b67s0bw7v8d3r-guile-2.0.
11
Dependencies:
  /gnu/store/bda98x7qxmd90qz4kr4rd03gbqsd038x-libunistring-0.9.3
  /gnu/store/j2ainmhcqcb5hgcqanywdzci84dns5lf-libtool-2.
4.2
  /gnu/store/lfgzbha6s711hynxdcj0xcd4l17582sy-libgc-7.4.
0
  /gnu/store/84xqp874r7a65alwvl11g6zrvk90812n-gmp-6.0.0a
debug     Install

Location  : gnu/packages/guile.scm:110:2
U:%*- *Guix Package Info* Top (1,0) (Guix-Info) 16:44
Quit
```

⇒ plan working **as expected**

⇒ plan working **as expected**

**embed DSLs!**



**Lively!**

# timeline

- ▶ July 2012 — GHM, Düsseldorf
- ▶ Nov. 2012 — dubbed GNU
- ▶ Jan. 2013 — 0.1
- ▶ June 2013 — European Lisp Symposium
- ▶ 27 Sep. 2013 — 0.4, with VM image
- ▶ Dec. 2013 — 0.5, system config, mips64
- ▶ Apr. 2014 — 0.6, signed binaries, guix system

# timeline

- ▶ July 2012 — GHM, Düsseldorf
- ▶ Nov. 2012 — dubbed GNU
- ▶ Jan. 2013 — 0.1
- ▶ June 2013 — European Lisp Symposium
- ▶ 27 Sep. 2013 — 0.4, with VM image
- ▶ Dec. 2013 — 0.5, system config, mips64
- ▶ Apr. 2014 — 0.6, signed binaries, guix system
- ▶ July 2014 — 0.7, **installable operating system!**

A white ceramic mug is filled to the brim with numerous small, round, brown pellets. The pellets have a slightly rough, textured surface. The word "status" is printed in a clean, white, sans-serif font across the center of the mug, partially overlapping the pellets. The background is a soft, out-of-focus green, suggesting an outdoor setting. The lighting is bright and even, highlighting the texture of the pellets and the smooth surface of the mug.

**status**

# status

- ▶ full-featured package manager
- ▶ 900 packages, 3 platforms
- ▶ **standalone distro!** (alpha)
- ▶ binaries built & served at <http://hydra.gnu.org>
- ▶ tooling: auto-update, sync descriptions with GNU, etc.
- ▶ I10n: 7 languages!

## Commits per Month



## In a Nutshell, GNU Guix...

... has had 3,403 commits made by 25 contributors representing 55,086 lines of code

... is mostly written in Scheme with a very well-commented source code

... has a young, but established codebase maintained by a large development team with stable Y-O-Y commits

... took an estimated 14 years of effort (COCOMO model) starting with its first commit in April, 2012 ending with its most recent commit 27 days ago

thanks for the code, reports, ideas, and translations!

- ▶ Eric Bavier, Taylan Ulrich Bayirli/Kammer, John Darrington, Eelco Dolstra & the Nix crew, Andreas Enge, Alírio Eyng, Guy Grant, Raimon Grau, Nikita Karetnikov, Aljosh Papsch, Manolis Ragkousis, Cyril Roelandt, Alex Sassmannshausen, Cyrill Schenkel, Jason Self, Sree Harsha Totakura, David Thompson, Mark H. Weaver
- ▶ Lluís Batlle i Rossell, Marek Benc, Daniel Clark, Alexandru Cojocaru, Aleix Conchillo Flaqué, Rafael Ferreira, Christian Grothoff, Jeffrin Jose, Kete, Alex Kost, Matthew Lien, Niels Möller, Yutaka Niibe, Adam Pribyl, Benno Schulenberg, Alen Skondro, Matthias Wachs, Zerwas
- ▶ Pavel Fric, Mario Blättermann, Felipe Castro, Balázs Úr, Rafael Ferreira, Miroslav Nikolic, Tran Ngoc Quân



## the road to 1.0

- 0. official FSDG compliance**
- 1. OS features: dm-crypt, LVM, etc.**
- 2. more service definitions**

## the road to 1.0

0. **official FSDG compliance**
1. **OS features:** dm-crypt, LVM, etc.
2. **more service definitions**
3. improved guix system reconfigure

## the road to 1.0

0. **official FSDG compliance**
1. **OS features**: dm-crypt, LVM, etc.
2. **more service definitions**
3. improved `guix system reconfigure`
4. **authenticated** `guix pull` (signed commits?)

## the road to 1.0

0. **official FSDG compliance**
1. **OS features:** dm-crypt, LVM, etc.
2. **more service definitions**
3. improved `guix system reconfigure`
4. **authenticated** `guix pull` (signed commits?)
5. **user interfaces:** Emacs, web, curses(?)

## the road to 1.0

0. **official FSDG compliance**
1. **OS features:** dm-crypt, LVM, etc.
2. **more service definitions**
3. improved guix system reconfigure
4. **authenticated** guix pull (signed commits?)
5. **user interfaces:** Emacs, web, curses(?)
6. larger, more robust **build farm**...
7. less dog food...
8. more packages...

## the road to 1.0

0. **official FSDG compliance**
1. **OS features:** dm-crypt, LVM, etc.
2. **more service definitions**
3. improved `guix system reconfigure`
4. **authenticated** `guix pull` (signed commits?)
5. **user interfaces:** Emacs, web, curses(?)
6. larger, more robust **build farm**...
7. less dog food...
8. more packages...
9. *your idea here*

*your help needed!*

- ▶ **install the distribution**
- ▶ **use it**, report bugs, add packages
- ▶ help with the **infrastructure** + admin
- ▶ share your **ideas!**

ludo@gnu.org



<http://gnu.org/software/guix/>



# credits

- ▶ GNU Guix logo, GFDL, <http://gnu.org/s/guix/graphics>
- ▶ NSA SIGDEV slide, <http://www.heise.de/ct/artikel/NSA-GCHQ-The-HACIENDA-Program-for-Internet-Colonization-229.html>
- ▶ dog food, [http://en.wikipedia.org/wiki/File:Kibbles\\_How\\_Others\\_See\\_Me.jpg](http://en.wikipedia.org/wiki/File:Kibbles_How_Others_See_Me.jpg)
- ▶ commit stats & project summary, <http://www.ohloh.net/p/gnuguix>

Copyright © 2010, 2012, 2013, 2014 Ludovic Courtès [ludo@gnu.org](mailto:ludo@gnu.org).  
Copyright of other images included in this document is held by their respective owners.

This work is licensed under the [Creative Commons Attribution-Share Alike 3.0](https://creativecommons.org/licenses/by-sa/3.0/) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

At your option, you may instead copy, distribute and/or modify this document under the terms of the [GNU Free Documentation License, Version 1.3 or any later version](https://www.gnu.org/licenses/gfdl.html) published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available at <http://www.gnu.org/licenses/gfdl.html>.

The source of this document is available from <http://git.sv.gnu.org/cgi/guix/maintenance.git>.