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 1,000 libraries with a few simple commands.

Quicklisp is easy to install and works with ABCL, Allegro CL, Clozure CL, CLISP, CMUCL, ECL, LispWorks, SBCL, and Scieneer CL, on Linux, Mac OS X, and Windows. The libraries were last updated on October 6, 2014.

To get started with the Quicklisp beta, download and loadhttp://beta.quicklisp.org/quicklisp.lisp

PGP signature
sha256 = 47cfdc0f4796b4c0ca05af16b666dd4b21c5b9e62b731390c895c2d19ed6d095

Quicklisp is provided as-is without warranty of any kind. See the release notes for known problems.

- Installation
- Loading After Installation
- Basic Commands
- Frequently Asked Questions
- License information
- Screencast

To get news and announcements, follow @quicklisp on twitter or read the Quicklisp blog. If you have any questions or comments, please email me, use the Quicklisp discussion group, or visit #quicklisp on freenode. Quicklisp code is hosted on GitHub.

Installation

To install Quicklisp, download quicklisp.lisp and load it.

To verify the integrity of quicklisp.lisp, you can download its detached PGP signature file and verify it against the Quicklisp release signing key, which has a fingerprint ofD7A3 489D DEFE 32B7 D0E7 CC61 3079 65AB 028B 5FF7, an id of 028B5FF7, and an email ofrelease@quicklisp.org.

You only need to load quicklisp.lisp once to install Quicklisp. See Loading After Installation for how to load the installed Quicklisp into your Lisp session after the initial installation.

Here's an example installation session on MacOS X with SBCL. In this example, **$** is the shell prompt and ***** is the Lisp REPL prompt.

```
$ curl -O http://beta.quicklisp.org/quicklisp.lisp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 49843  100 49843    0     0  33639      0  0:00:01  0:00:01 --:--:-- 50397
```

```
$ curl -O http://beta.quicklisp.org/quicklisp.lisp.asc
...
$ gpg --verify quicklisp.lisp.asc
gpg: Signature made Sat Feb  1 09:25:28 2014 EST using RSA key ID 028B5FF7
gpg: Good signature from "Quicklisp Release Signing Key "
$ sbcl --load quicklisp.lisp
This is SBCL 1.0.42.52, an implementation of ANSI Common Lisp.
More information about SBCL is available at <http://www.sbcl.org/>.

SBCL is free software, provided as is, with absolutely no warranty.
It is mostly in the public domain; some portions are provided under
BSD-style licenses.  See the CREDITS and COPYING files in the
distribution for more information.

  ==== quicklisp quickstart loaded ====

    To continue, evaluate: (quicklisp-quickstart:install)

* (quicklisp-quickstart:install)

; Fetching #<URL "http://beta.quicklisp.org/quickstart/asdf.lisp">
; 144.48KB
==================================================
147,949 bytes in 0.64 seconds (226.11KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/quickstart/quicklisp.tar">
; 160.00KB
==================================================
163,840 bytes in 0.76 seconds (211.36KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/quickstart/setup.lisp">
; 2.78KB
==================================================
2,846 bytes in 0.001 seconds (2779.30KB/sec)
Upgrading ASDF package from version 2.004 to version 2.009
; Fetching #<URL "http://beta.quicklisp.org/dist/quicklisp.txt">
; 0.40KB
==================================================
408 bytes in 0.003 seconds (132.81KB/sec)

  ==== quicklisp installed ====

    To load a system, use: (ql:quickload "system-name")

    To find systems, use: (ql:system-apropos "term")

    To load Quicklisp every time you start Lisp, use: (ql:add-to-init-file)
    For more information, see http://www.quicklisp.org/beta/

NIL
* (ql:system-apropos "vecto")

; Fetching #<URL "http://beta.quicklisp.org/dist/quicklisp/2010-10-07/systems.txt">
; 45.30KB
==================================================
46,386 bytes in 0.50 seconds (89.70KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/dist/quicklisp/2010-10-07/releases.txt">
; 83.49KB
==================================================
85,490 bytes in 0.53 seconds (157.22KB/sec)
#<SYSTEM cl-vectors / cl-vectors-20101006-git / quicklisp 2010-10-07>
#<SYSTEM lispbuilder-sdl-cl-vectors / lispbuilder-20101006-svn / quicklisp 2010-10-07>
#<SYSTEM lispbuilder-sdl-cl-vectors-examples / lispbuilder-20101006-svn / quicklisp 2010-10-07>
#<SYSTEM lispbuilder-sdl-vecto / lispbuilder-20101006-svn / quicklisp 2010-10-07>
#<SYSTEM lispbuilder-sdl-vecto-examples / lispbuilder-20101006-svn / quicklisp 2010-10-07>
#<SYSTEM static-vectors / static-vectors-20101006-git / quicklisp 2010-10-07>
#<SYSTEM vecto / vecto-1.4.3 / quicklisp 2010-10-07>
NIL
* (ql:quickload "vecto")
To load "vecto":
  Install 5 Quicklisp releases:
    cl-vectors salza2 vecto zpb-ttf zpng
; Fetching #<URL "http://beta.quicklisp.org/archive/salza2/2010-10-06/salza2-2.0.7.tgz">
; 14.84KB
==================================================
15,197 bytes in 0.12 seconds (125.77KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/archive/zpng/2010-10-06/zpng-1.2.tgz">
; 38.59KB
==================================================
39,521 bytes in 0.37 seconds (103.47KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/archive/zpb-ttf/2010-10-06/zpb-ttf-1.0.tgz">
; 42.59KB
==================================================
43,611 bytes in 0.39 seconds (110.33KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/archive/cl-vectors/2010-10-06/cl-vectors-20101006-git.tgz">
; 40.40KB
==================================================
41,374 bytes in 0.37 seconds (109.20KB/sec)
; Fetching #<URL "http://beta.quicklisp.org/archive/vecto/2010-10-06/vecto-1.4.3.tgz">
; 75.71KB
==================================================
77,526 bytes in 0.49 seconds (153.57KB/sec)
```

```
; Loading "vecto"
..................................................
[package zpb-ttf].................................
[package salza2]..................................
[package zpng]....................................
[package net.tuxee.paths].........................
[package net.tuxee.aa]............................
[package net.tuxee.aa-bin]........................
[package net.tuxee.vectors].......................
[package vecto].......
("vecto")
* (ql:add-to-init-file)
I will append the following lines to #P"/Users/quicklisp/.sbclrc":

  ;;; The following lines added by ql:add-to-init-file:
  #-quicklisp
  (let ((quicklisp-init (merge-pathnames "quicklisp/setup.lisp"
                                         (user-homedir-pathname))))
    (when (probe-file quicklisp-init)
      (load quicklisp-init)))

Press Enter to continue.


#P"/Users/quicklisp/.sbclrc"
* (quit)
$
```

Although this example uses a Unix shell, the same procedure would work for installation in the REPL of a Windows or Mac GUI-based CL implementation.

Loading After Installation

You only need to install Quicklisp with `quicklisp.lisp` once. To load Quicklisp into your Common Lisp session after the initial installation, load the file`quicklisp/setup.lisp` with the Common Lisp `load` function.

For example, if you installed Quicklisp in the default location in your home directory, the following command suffices to load Quicklisp into a running session:

```
(load "~/quicklisp/setup.lisp")
```

The path must be adjusted if Quicklisp is loaded in a different location, or if your implementation does not support the ~ pathname syntax (i.e. CMUCL).

The function `ql:add-to-init-file` will add code to do this to your init file automatically, so Quicklisp will load whenever your Common Lisp session starts.

Basic Commands

To load software, use:

```
(ql:quickload system-name)
```

For example:

```
CL-USER> (ql:quickload "vecto")
To load "vecto":
  Install 5 Quicklisp releases:
    cl-vectors salza2 vecto zpb-ttf zpng
loading output here
```

Quicklisp will automatically download any supporting software it needs to load the system.

Please note that some systems have different names than their projects. For example, to load cl-yacc's system, run `(ql:quickload "yacc")`, not `(ql:quickload "cl-yacc")`.

`ql:quickload` can also be used to load local systems that aren't part of Quicklisp. If the local system depends on software that is available through Quicklisp, the dependencies will be automatically loaded.

By default, `ql:quickload` hides most compilation and loading output, including warnings, and shows progess as a series of dots. You can show full compilation and loading output by passing `:verbose t` as arguments to `ql:quickload`. This output can be especially helpful when reporting and troubleshooting problems.

To remove software, use:

```
(ql:uninstall system-name)
```

For example:

```
CL-USER> (ql:uninstall "vecto")
T
```

An uninstall does the following:

- Deletes the system's tarball archive and unpacked source files
- Deletes Quicklisp metadata files associated with the system
- Clears ASDF's system cache via `asdf:clear-system`

Uninstalling does *not* alter the current Lisp session in any other way; if the given system has been loaded, it remains loaded and accessible until the session is ended.

To find out what's available in Quicklisp, use:

```
(ql:system-apropos substring)
```

For example:

```
CL-USER> (ql:system-apropos "xml")
#<SYSTEM bknr.xml / bknr-datastore-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cl-libxml2 / cl-libxml2-0.3.4 / quicklisp 2010-09-16>
#<SYSTEM cxml / cxml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cxml-dom / cxml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cxml-klacks / cxml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cxml-rng / cxml-rng-2008-11-30 / quicklisp 2010-09-16>
#<SYSTEM cxml-rpc / cxml-rpc-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cxml-stp / cxml-stp-2008-11-30 / quicklisp 2010-09-16>
#<SYSTEM cxml-stp-test / cxml-stp-2008-11-30 / quicklisp 2010-09-16>
#<SYSTEM cxml-test / cxml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM cxml-xml / cxml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM pithy-xml / pithy-xml-20100916-git / quicklisp 2010-09-16>
#<SYSTEM s-xml / s-xml-20100916-http / quicklisp 2010-09-16>
#<SYSTEM s-xml-rpc / s-xml-rpc-20100916-http / quicklisp 2010-09-16>
#<SYSTEM xml-render / cl-typesetting-20100916-svn / quicklisp 2010-09-16>
#<SYSTEM xmls / xmls-1.3 / quicklisp 2010-09-16>
```

The system name is the word after "SYSTEM" and before the first "/", e.g. "cxml-stp". System names are used by `ql:quickload` and for expressing dependency information in system definitions.

While `ql:system-apropos` is useful for simple string searches of system and project names, Quickdocs provides a more general full-text search for systems and projects.

To load Quicklisp when you start Lisp, use:

```
(ql:add-to-init-file)
```

Quicklisp will append code to your Lisp's init file that will load Quicklisp on startup.

To install and configure SLIME, use:

```
(ql:quickload "quicklisp-slime-helper")
```

Then follow the directions it displays. `quicklisp-slime-helper` will create a file you can load in Emacs that configures the right load-path for loading Quicklisp's installation of SLIME.

To get updated software, use:

```
(ql:update-all-dists)
```

To update the Quicklisp client, use:

```
(ql:update-client)
```

To go back to a previous set of libraries, see the blog post Going back in (dist) time. This can be useful if your project depends on the state (APIs, etc) of the Quicklisp library universe from a particular point in time.

To see what systems depend on a particular system, use:

```
(ql:who-depends-on system-name)
```

License information

```
Copyright © 2010 Zachary Beane <zach@quicklisp.org>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

Portions of Quicklisp are derived from Pierre Mai's Deflate library, which is available under the following terms:

```
Deflate --- RFC 1951 Deflate Decompression

Copyright (C) 2000-2009 PMSF IT Consulting Pierre R. Mai.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
```

Zach Beane 2011-06-19