

# Registry Installation - Source

- [Download Source](#)
  - [Downloading the Latest Release](#)
  - [Downloading the Latest Master](#)
  - [Cloning the Git Repo](#)
- [Configure Web Server](#)
- [Verify File Permissions](#)
- [Create /tmp Directory](#)
- [Integrate Web Server Authentication](#)
- [Integrate Web Server Logout](#)
- [Next: Registry Installation - Database](#)

## Download Source

### Downloading the Latest Release

In general, you should download the latest release. This will make it easier to track where your deployment is versus the current development work for purposes of reporting bugs, diagnosing issues, and understanding available features.

Download the COnanage Registry source files somewhere into the file system. The location you put the files does not have to be the location from which the files are served by the web server. Create a symlink to the directory called `registry-current` (or something similar):

```
$ mkdir /srv/comanage
$ cd /srv/comanage
$ wget https://github.com/Internet2/comanage-registry/archive/4.3.2.tar.gz
$ tar xzf 4.3.2.tar.gz
$ ln -s comanage-registry-4.3.2 registry-current
```

### Downloading the Latest Master

The `master` branch has the latest features that should be stable enough for use, but may not be feature complete. (Typically, though, `master` is the same as the latest release.) If you want the latest master, you can download it instead:

```
$ wget https://github.com/Internet2/comanage-registry/archive/master.tar.gz
```

You can also download the `develop` branch, which is usually more bleeding edge.

```
$ wget https://github.com/Internet2/comanage-registry/archive/develop.tar.gz
```

### Cloning the Git Repo

If you plan on mucking around with the code, you can also [clone the git repo](#).

## Configure Web Server

Deploy the COnanage Registry directory wherever you like. Note that the user that the web server runs as needs to be able to read all the files.

Configure your web server to deliver the registry at a suitable URL such as <https://some-vo.org/registry>. A simple strategy to accomplish this when running under the Apache web server is to create a symlink in the `DocumentRoot` named `registry` that points to the directory `.../registry-current/app/webroot`:

```
$ cd /var/www/html
$ ln -s /path/to/registry-current/app/webroot registry
```

The preferred path `/var/www` or `/var/www/html` may vary according to your operating system, distribution, or web server configuration.



### Installation at /registry Currently Recommended

As of Registry v4.0.0, it is no longer required to install at the URL path `/registry`. However, it is recommended to use this path unless a specific reason requires the use of a different path.

Versions prior to v4.0.0 require CManage Registry to be made available at <https://your-site.org/registry>.

## Verify File Permissions

You should verify that the web server will not deliver unprocessed files, especially configuration files such as the database configuration file (ie: <https://some-vo.org/registry/app/Config/database.php>). By default, these files will not be delivered.

## Create /tmp Directory

As of v1.0.0, `app/tmp` is a symlink to `local/tmp`. You'll most likely want to make that a symlink to another location, since it is bad practice to have writable directories on the file system delivering web content. A reasonable alternative would be `/var/cache/registry`. The easiest way to do this on a Unix-like system is to create a symlink to the new directory.

The basic required structure for the `tmp` directory is included in `app/tmp.dist`. Be sure to replicate this in the target location.

```
$ cd registry-current/app
$ sudo cp -r tmp.dist /var/cache/registry
$ sudo chown -R $HTTPUSER /var/cache/registry
$ sudo chmod 700 /var/cache/registry
$ cd registry-current/local
$ ln -s /var/cache/registry tmp
```



### Log Files May Be Written To The Cache Directory

The CakePHP framework may write error and debugging logs to the `logs` directory under the `tmp` directory. You may wish to monitor and/or rotate these files. By default, the framework usually rotates the log files when they get large.

Note you can point the subdirectories of `tmp` to different locations. For example, you could point `tmp/logs` to `/var/log/registry` if you want to keep all of your logfiles in the same place.



For versions prior to v1.0.0, update `app/tmp` to point directly to your `tmp` directory.

## Integrate Web Server Authentication

In order to integrate CManage Registry with your authentication system, configure your Web server to protect the directory `registry/app/webroot/auth/login`. For example, under Apache your configuration may look something like

```
DocumentRoot /var/www
<Directory /var/www/registry/auth/login>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    Require shib-session
</Directory>
<Location / >
    AuthType shibboleth
    Require shibboleth
</Location>
```

For Apache, it is recommended that you place this configuration in `httpd.conf` rather than `.htaccess`, to simplify future upgrades.

If you choose to use a SAML2 service provider (SP) such as the Shibboleth Native SP or SimpleSAMLPhP for authentication you may find the [CManage Registry Shibboleth Embedded Discovery Service Plugin](#) useful.

## Integrate Web Server Logout

COmanage Registry uses a standard PHP session to track requests per user. Clicking the "Logout" button ends the PHP session but does not end any other session such as a session created during authentication by the web server authentication mechanism or module such as the Shibboleth Native SP for Apache (Shibboleth) or mod\_auth\_openidc. The web server authentication mechanism should be configured so that any necessary session termination happens when the browser accesses `registry/auth/logout`.

For example with Shibboleth the following configuration will terminate the Shibboleth session and then allow the browser to continue to access `registry/auth/logout`:

```
RewriteEngine On
RewriteCond %{QUERY_STRING} !after_redirect
RewriteRule ^/registry/auth/logout.* https://%{SERVER_NAME}/Shibboleth.sso/Logout?return=https://%{SERVER_NAME}
/registry/auth/logout/?after_redirect [L,R]
```

**Next: [Registry Installation - Database](#)**