Technical Documentation : Drupal project and CVS notes

This page last changed on 2009-01-14 by todd.

Logging in to CVS

- Open Terminal (or any command-line program).
- Go to your local Drupal repository folder (in this case, *~/drupal*).

```
cd drupal
```

- Run the following commands to log in (replace "YOURUSERNAME" with your CVS username):

```
export CVSROOT=:pserver:YOURUSERNAME@cvs.drupal.org:/cvs/drupal-contrib
cvs login
```

- You will be prompted for your password.
- After logging in, you will need to check out the root */contributions/modules* directory:

```
cvs co -l contributions/modules
cd contributions/modules
```

Creating a new CVS project

Culled from Drupal.org's guide: Step-by-step: Create a CVS project.

Committing a new project to CVS

- Follow the login instructions above.
- Move your project files into the current directory (*~/drupal/contributions/modules*):

```
mv PATH/TO/MODULENAME .
```

- Add your files to the repository:

```
cvs add MODULENAME
cd MODULENAME
cvs add *
cvs commit -m "MESSAGE"
```

Note: After completing these steps above, you've simply committed your new project to Drupal's CVS repository in the default HEAD branch. Your must create a new project on Drupal.org and tag it for release before anyone can download it.

Creating a new project on Drupal.org

- Visit the Submit project page
- Make sure that the "Short project name" matches the directory name in the CVS repository. For example, the *contributions/modules/my_module* module has the short name my_module.
- Fill in the correct value for the CVS directory field to match the path in the repository. In this case, you should include everything after the contributions part of the directory name: */modules/my_module/*

Creating the first branch of the project

The following information was culled from Branches on Drupal.org.

It's highly suggested that you branch your project immediately after committing it to CVS and creating the project on Drupal.org. By default, newly committed projects are filed into the HEAD branch, which is typically reserved for bleeding-edge development – that is, modules that are compatible with the latest *development* release of Drupal, not standard releases.

The first branch should correspond to the version of Drupal your project was designed for. See the instructions below on how to branch projects.

Branching projects

Branch names

What follows is a highly condensed version of Drupal.org's Overview of contributions branches and tags document.

First, you need to understand the meaning behind branch names. (Unless your project has major releases – say, 1.x and 2.x – you can disregard the "Version 2" items below.)

- Drupal 5.x
  - Version 1 (5.x-1.x): DRUPAL-5
  - Version 2 (5.x-2.x): DRUPAL-5--2
- Drupal 6.x
  - Version 1 (6.x-1.x): DRUPAL-6--1
  - Version 2 (6.x-2.x): DRUPAL-6--2

It's important that you note the different between Drupal 5.x and 6.x above: 5.x does **not** require a *--1* after *DRUPAL-5*. However, you will need to add *--2* if you create a 2.x version of the project.

Why branch?

Branching is necessary to achieve two goals:

1. Maintain projects with more than one Drupal core compatibility. In other words, you need to branch your project if you want to maintain versions that work with either Drupal 5.x or 6.x.
2. Create automatic development snapshots of your commits on Drupal.org. (More on this later...)

Making a new branch

Branching code is exactly like tagging a release, which is described in detail later on. The exception, however, is the use of the *-b* operator, which tells CVS to create a branch instead of a tag.

- Follow the login instructions above.
  - OR, if you're creating your first release of a new project, make sure that you've already done the following:
    - Committed your project to CVS
    - Created a project on Drupal.org
- Branch the project according to its core compatibility (Drupal 5.x or 6.x) and major version (1.x, 2.x, etc.):
  - Drupal 5.x-1.x:

```
cvs tag -b DRUPAL-5 path/to/my_module
```

- Drupal 5.x-2.x:

```
cvs tag -b DRUPAL-5--2 path/to/my_module
```

- Drupal 6.x-1.x:

```
cvs tag -b DRUPAL-6--1 path/to/my_module
```

- Drupal 6.x-2.x:

```
cvs tag -b DRUPAL-6--2 path/to/my_module
```

Creating a new module release

Before continuing, you should check out Lullabot's excellent screencast Create a Module Release.

Creating new module release involves "tagging" the files using a string that corresponds to its version number. To create a new release, you must "tag" it.

Before we tag any new releases, however, we need to check out the project from CVS.

Checking out the latest version of a branch

The following was culled from Drupal.org's Checking out from the contributions repository.

- Follow the login instructions above.
  - OR, if you're creating your first release of a new project, make sure that you've already done the following:
    - Committed your project to CVS
    - Created a project on Drupal.org
    - Branched the project according to its core compatibility with Drupal (5.x or 6.x)
- Check out the branch of the module you want to upgrade.
  - Drupal 5:

```
cvs co -r DRUPAL-5 -d local/path/to/my_module_5 contributions/modules/my_module
```

    Here, "DRUPAL-5" refers to the Drupal 5 branch of the module.
  - Drupal 6:

```
cvs co -r DRUPAL-6--1 -d local/path/to/my_module_6 contributions/modules/my_module
```

Before continuing, let's break down those commands.

Command	Meaning
`cvs co`	Check out the code...
`-r DRUPAL-5`	...from branch Drupal 5.x-1.x (this is the name of a branch, not a tag)...
`-d local/path/to/my_module_5`	...into my local directory *local/path/to/my_module_5* (I prefer splitting branches into their own directories to avoid confusion)...
`contributions/modules/my_module`	...of project "my_module".

Adding a new tag

Now that you have the most recent version of your project, you can make changes to the code. It's easiest to develop a new release on a development instance and simply overwrite the files in your local Drupal repository. It's possible, however, that other developers have committed changes to the repository without your knowledge, so you should probably double-check the code you just pulled from CVS.

- Go to the directory that contains your project:

```
cd local/path/to/my_module_5
```

- Add a tag:

```
cvs tag DRUPAL-5--1-0-RC
```

Note: The *-RC* stands for "release candidate" and is not required. You can use *-RC1*, *-RC2*, *-ALPHA1*, *-BETA*, etc. Be advised, however, that "beta" typically flags a project as not yet ready for public release. "Release candidate" is preferred for more mature releases. For more information about tag naming, see Drupal.org's Overview of contributions branches and tags document.

Removing a tag

If you messed up, don't worry. You can remove a tag using the *-d* operator:

```
cvs tag -d DRUPAL-5--1-0
```

Create a release on Drupal.org

...