# Developers Guide

***0.2***

WIP
3/2/2011



## Table of Contents





This section will show you how to setup a razorback development environment.

## *Dependencies*

As of version 0.1.1 the following packages are required to be installed:

- Ubuntu (10.10)
  - libmysqlclient-dev
  - uuid-dev
  - libmagic-dev
  - autoconf
  - libssl-dev
  - libpcre3-dev
- FreeBSD
  - devel/autoconf
  - devel/automake
  - devel/e2fsprogs-libuuid
  - devel/libtool
  - devel/pcre
  - devel/gmake
  - database/mysql51-client

You will also need a MySQL server to run the database on.

## *Getting the code*

You have two options for getting the code you can check out a single meta project with all of the components in it or you can check out the individual components that you wish to work on.



You can check out the latest development meta project version using the following command:

```
svn co https://razorbacktm.svn.sourceforge.net/svnroot/razorbacktm/releases/trunk razorback
```

Alternatively you can checkout the API and dispatcher and some nuggets separately:

```
mkdir razorback
```

```
cd razorback
svn co https://razorbacktm.svn.sourceforge.net/svnroot/razorbacktm/api/trunk api
svn co
https://razorbacktm.svn.sourceforge.net/svnroot/razorbacktm/dispatcher/trunk  \
dispatcher
svn co \

https://razorbacktm.svn.sourceforge.net/svnroot/razorbacktm/nuggets/masterNugge
t/trunk \
 masterNugget

svn co \

https://razorbacktm.svn.sourceforge.net/svnroot/razorbacktm/nuggets/smtpParser/tr
unk \
 smtpParser
```

## *Setting up the database*

On the database server you need to do the following:

- Create the database
- Load the schema
- Grant the permissions.

The following example assumes knowledge of MySQL:

```
$ cd razorback
$ mysql -uroot -p
Enter password:
mysql> create database razorback2;
Query OK, 1 row affected (0.00 sec)
mysql> grant all on razorback2.* to razorback2@localhost identified by
‘razorback2’;
Query OK, 0 rows affected (0.00 sec)
mysql> use razorback2;
Database changed
mysql> \. dispatcher/share/razorback.sql
.....
mysql> \q
Bye
$
```



## *Building the system*

It is recommended that you create a dedicate area to install the development versions of the applications and libraries so that if things go wrong you can clean up the environment and start a fresh very easily. In the following examples I will be using ~/razorback-install.

Building the meta project is as simple as:

```
$ export PREFIX=~/razorback-install
$ mkdir $PREFIX
$ cd ~/razorback
$ ./configure –prefix=$PREFIX
$ make && make install
```

You can use the following process to build the system when using checkouts of the individual components.

```
$ export PREFIX=~/razorback-install
$ mkdir $PREFIX
$ for I in api dispatcher masterNugget smtpParser; do
        (
        cd ~/razorback/$I
        ./configure –prefix=$PREFIX
       make && make install
       );
       done
```

## *Naming Conventions*

Nugget Naming

Each nugget should be named after the functionality it provides. If the name has more than one word in it then the names should be concatenated together capitalizing the first letter of each word from the second word onwards. For example the master nugget is called masterNugget and the file injection nugget is called fileInject.



## *Release Components*

Each Razorback™ release for public consumption comes in two forms, the combined single tarball release, and the individual components that make up the system are released individually.



## *Repository Structure*

- api
  - Trunk
  - branches
    - 0.1
  - tags
    - 0.1-release
    - 0.1.1-release
- dispatcher
  - trunk
  - branches
    - 0.1
  - tags
    - 0.1-release
    - 0.1.1 – release
- common
  - trunk
  - branches
    - 0.1
  - tags
    - 0.1-release
    - 0.1.1 – release
- nuggets
  - nug_1
    - trunk
    - branches
      - 0.1
    - tags
      - 0.1-release
  - nug_2
    - trunk
    - branches
      - 0.1
    - tags
      - 0.1-release
- doc
  - trunk
  - branches
    - 0.1
  - tags
    - 0.1-release
    - 0.1.1 – release
- releases
  - trunk
  - 0.1
  - 0.1.1



## *What are all these things?*

Razorback™ consits of the following sub projects:

- The *api* containing the shared API for nuggets and dispatcher.
- The common containing the autotools and other code that are shared between the projects.
- The *dispatcher* containing the dispatcher code.
- Each *nugget* exists as a separate project in the nuggets directory.
- The *doc* project contains top-level documentation for the system.

The releases directory is used as a collection of Meta projects used to pull in the required projects for a release or build environment. These projects will be used to create the single tarball releases with all the required components in them to get Razorback™ up and running quickly.

## *Project layout*

Each project is laid out with trunk, branches and tags directories for tracking development in typical subversion development model which is also detailed [in the Subversion book.]

Each projects development area will be laid out as a standard autotools project as [described in the autotools book.]

## *Release Meta project layout*

The following layout is for the development image rooted at /releases/trunk

- configure (local)
- api (extern:/api/trunk)
- dispatcher (extern:/dispatcher/trunk)
- doc (extern:/doc/trunk)
- nuggets (local)
    - nug_1 (extern:/nuggets/nug_1/trunk)
    - nug_2 (extern:/nuggets/nug_2/trunk)



## *Why did we choose this layout?*

This layout was chosen to future proof the project so that in the long run each component can be packaged up individually for the various OS package management systems, allowing users to install larger network of nuggets and collectors without having to install the whole system on every box.