Dexy Install Guide

Ana Nelson

November 14, 2010

Before you begin, you should make sure you have the most recent version of this document. You can obtain this from the bitbucket cache or you can check out the source and compile it yourself.

Visit `http://dexy.it` for more information about Dexy.

Introduction

Right now, the recommended way to install and run Dexy is within an Ubuntu Lucid virtual machine, and these instructions describe in detail how to do this. If you would like to install Dexy on your own machine or a non-Ubuntu virtual machine, then these instructions will help you to understand what needs to be installed and in what order, and you can adjust the actual commands as needed.

If you are an experienced user, and would like to just work through (or run) an install script, then go here. The script is designed especially to run on the Turnkey Linux Lucid Base ISO.

This script is run as sudo, although not all steps require this. You will want to change file ownership after running this script as all downloaded files will be owned by root (will tidy this up soon).

Setup

At the beginning of the script, we define a few constants which we will use later on. You should change these to vaules that make sense for you.

```
# Please change these constants to values appropriate for you.

# The CRAN_MIRROR should be your local mirror from this list
# http://cran.r-project.org/mirrors.html
CRAN_MIRROR="http://ftp.heanet.ie/mirrors/cran.r-project.org"

# A bash profile config file, we are going to append the PATH
# with the Ruby gem bin directory so we have RedCloth.
BASH_PROFILE="/home/ana/.bashrc"
```

To keep things tidy, we create a directory to hold various files we will be downloading.

```
mkdir install-src
cd install-src
```

Next we tell Ubuntu to enable the multiverse, which gives us access to more packages.

```
echo "deb http://archive.ubuntu.com/ubuntu lucid multiverse" >> /etc/apt/sources.list.d/sources.list
```

Before we start installing software, and especially because we have added the multiverse, we want to update package definitions.

```
apt-get update
```

Next we install some general utilities which will help us in later installations.

```
apt-get install -y apt-utils
apt-get install -y build-essential
apt-get install -y python-dev
apt-get install -y pkg-config
```

And some common version control systems so we can pull in external repositories.

```
apt-get install -y mercurial
apt-get install -y subversion
```

We install Python's setuptools as we will need this to unstall Dexy from source, and it also gives us access to easy_install which we will use in a few places.

```
apt-get install -y python-setuptools
```

And then we install a full version of vim, since this will be useful when we are working with Dexy.

```
apt-get install -y vim
```

Dexy

Now we're ready to start installing Dexy! First we install some packages which Dexy depends upon.

```
apt-get install -y python-pydot
apt-get install -y python-simplejson
apt-get install -y python-nose
easy_install ordereddict
easy_install ansi2html
```

Then a few packages which aren't strictly necessary to install Dexy, but which are depended upon by many of Dexy's filters.

```
apt-get install -y python-pexpect
apt-get install -y python-jinja2
easy_install http://dexy.it/tmp/idiopidae-0.5.tgz
easy_install http://dexy.it/tmp/zapps-0.5.tgz
```

Then some Python libraries for working with HTML and CSS (useful in our web server and also in some of the Dexy filters).

```
easy_install pygments
easy_install BeautifulSoup
easy_install cssutils
easy_install pynliner
```

Now we get Dexy's source code out of the Mercurial repository (we installed Mercurial earlier, remember?) and install Dexy!

```
hg clone http://bitbucket.org/ananelson/dexy
cd dexy
python setup.py install
cd ..
```

If you think you'll be hacking on Dexy you might want to change 'install' to 'develop'.

Okay, that's the basic Dexy install. You can stop here if you want, but now we're going to install more software to make Dexy actually useful and to be able to run all the Dexy examples.

Web Server

Let's install software for the built-in web server next. First, some preliminaries.

```
apt-get install -y sqlite3 libsqlite3-dev
apt-get install -y uuid-dev
```

Then ZeroMQ

```
wget http://www.zeromq.org/local--files/area:download/zeromq-2.0.10.tar.gz
tar -xzvf zeromq-2.0.10.tar.gz
cd zeromq-2.0.10
./configure
make
make install
cd ..
```

and pyzmq (the Python interface to ZeroMQ)

```
ldconfig
sudo easy_install pyzmq
```

Then we install the mongrel2 server (mongrel2 uses ZeroMQ)

```
wget http://mongrel2.org/static/downloads/mongrel2-1.4.tar.bz2
tar -xjvf mongrel2-1.4.tar.bz2
cd mongrel2-1.4
make install
```

and the Python helper libraries for mongrel2

```
cd examples/python
python setup.py install
cd ../../..
```

That's everything installed. A mongrel2 server works in 2 parts, we're going to start running 1 part now, and the other part when we're finished installing everything. You could also wait until the end to start the server (and there's no need to use the web server part of Dexy at all if you don't want to).

```
cd dexy/servers/live-server
mkdir logs
mkdir run
mkdir tmp
m2sh start -uuid 7daa8d71-c5f5-4a29-b027-14ccfb5bd6a5
cd ../../../
```

R and LaTeX

Now we're going to start installing software to run the Dexy examples. If you don't use R or LaTeXyou can skip this part, of course you won't be able to run any examples that depend upon them.

Before we can install R, we need to obtain the public key of the package publisher

```
apt-get install -y gnupg
gpg --keyserver keyserver.noreply.org --recv-key E2A11821
gpg -a --export E2A11821 | sudo apt-key add -
```

Then we add CRAN's Ubuntu repository to our sources, update, and finally we can install R.

```
echo "deb $CRAN_MIRROR/bin/linux/ubuntu lucid/" >> /etc/apt/sources.list.d/sources.list
apt-get update
apt-get install -y r-base-dev
```

Then we can start to install some R packages.

```
echo "install.packages(\"rjson\", repos=\"$CRAN_MIRROR\")" | R --vanilla
```

Next we install LaTeX. The full LaTeXinstall can take quite a long time, and it's a very large file to download. Unless you have time or space limitations, I would suggest you download it as it means you don't have to spend time later hunting down missing packages, but if you don't use LaTeXand don't care about running the Dexy examples then by all means leave this out (you'll have a much smaller image if you do).

```
apt-get install -y texlive-full
```

Examples

Now we install software which happens to be used for the various examples included with Dexy.

```
apt-get install -y clang
apt-get install -y ragel
apt-get install -y espeak
apt-get install -y lame
```

We install Ruby, rubygems and the RedCloth gem in particular.

```
apt-get install -y rubygems
apt-get install -y ruby1.8-dev
gem install RedCloth
echo "PATH=\"/var/lib/gems/1.8/bin:$PATH\"" >> $BASH_PROFILE
```

GarlicSim, a Python simulation framework

```
easy_install garlicsim
easy_install garlicsim-lib
```

Several packages to enable us to install the seewave package in R (which, yes, I am starting to think may have been a bad choice for my first example!)

```
apt-get install -y fftw3 fftw3-dev
apt-get install -y xorg-dev
apt-get install -y freeglut3-dev
echo "install.packages(\"fftw\", repos=\"$CRAN_MIRROR\")" | R --vanilla
echo "install.packages(\"rgl\", repos=\"$CRAN_MIRROR\")" | R --vanilla
echo "install.packages(\"seewave\", repos=\"$CRAN_MIRROR\")" | R --vanilla
```

Now, at last, we can download the example repository.

```
hg clone http://bitbucket.org/ananelson/dexy-examples
cd dexy-examples
```

```
mkdir artifacts
mkdir logs
dexy -p .
```

A few of the examples expect to find a stylesheet here, so we generate one. (We could also commit this file in the repo, but this is a handy reminder of the pygmentize syntax.)

To get started, we run a few examples (clearing out the cache the first time).

```
hg clone http://bitbucket.org/ananelson/dexy-examples
cd dexy-examples
mkdir artifacts
mkdir logs
dexy -p .
```

Then, we start the second part of our web server. This is the handler which we can point at any Dexy project directory. In this case, we're going to point it at the examples directory.

```
dexy-live-server .
```

Now, you should have a server running on port 6767 of the virtual machine. If you are using the TurnKey Linux ISO, then the configuration screen will display the IP address, and you should be able to navigate to this IP address and port and see a listing of the examples in the directory.