Source Repositories

OpenIndiana maintains source repositories for the included software. Some repositories are direct clones of upstream repositories opensolaris.org or genunix.org, others are specific to OpenIndiana.

They are in the form of Mercurial repositories can be cloned via HTTP and modified over SSH.

You can browse all of the repositories here: http://hg.openindiana.org/.
Specify the HTTP URL of a repository to `hg clone` to create a clone.

See Building the Operating System for more details about the repository layout.

Information for repository maintainers

Configuring the web server

Create /etc/apache2/2.2/conf.d/pkgdev.conf:

```
<VirtualHost *:80>
ServerName pkgdev.openindiana.org

DocumentRoot /var/virtualhosts/pkgdev.openindiana.org

<Directory "/usr/demo/mercurial">
AllowOverride None
Options ExecCGI
Order allow,deny
Allow from all
</Directory>

ScriptAlias /hg /usr/demo/mercurial/hgwebdir.fcgi

</VirtualHost>
```

Cloning a Mercurial repo

Log in to pkgdev.openindiana.org. Create a filesystem for the new repo under data/export, named as the repo:

```
pfexec zfs create data/export/wombat
```

Change it to owned by the hg user, hg group:

```
pfexec chown hg:hg /data/export/wombat
```

Tell the Mercurial web app about it, by adding a line to /usr/demo/mercurial/hgweb.config:

```
wombat = /data/export/wombat
```

Switch to the hg user, and clone into the filesystem:

```
cd /data/export
hg clone -U ssh://anon@hg.opensolaris.org/hg/wombat/source wombat
```

(If you forget the -U flag, do an hg update null in the repo after the clone.) Edit the repository's web information, /data/export/wombat/.hg/hgrc:

```
[web]
description = Code for all wombats
```

The hg user has a cron job which runs a script (update-repos.sh) which pull updates from remote repos. Add this repo to the script.

Cloning a Subversion repo

This is a bit fiddlier, as it needs to use a custom-built copy of hg. But the only step that's really different is the cloning. Do this instead:

```
cd /data/export
export PATH=/data/export/bin:$PATH
export PYTHONPATH=/data/export/lib/python/
hg convert <your subversion url here> wombat
```

Converting big subversion repositories is seriously slow, consider running hg inside screen or using nohup.

The cron job can't update a converted repo. You basically have to re-run the conversion step again - but it is smart enough to do an incremental conversion.

Pushing changes to a repo

We allow a number of users to push changes to the repos. We use the model of a single "hg" account which can only be accessed via ssh with public keys, and the authorized committers have their public keys in the hg account's .ssh/authorized_keys file.