How to Create Local Debian Repository

Why to create a private Debian repository ?

So that to gain more control while updating and upgrading your `.deb` files, to be able to install exactly the files you used in your test environment, and, what is also important – to avoid so called “dependency hell”.

Follow the steps below to create a local Debian repository, that is recognized by Apt. Apt will use .deb files from the private repository and resolve the associated dependencies when you run the command to install the packages.

1. Install the dpkg-scanpackages tool

To do so, run the following command as root:

```
apt-get install dpkg-dev
```

2. Then, create repository directory and place packages

there:

```
mkdir /path/to/repository
cp /path/to/packages /path/to/repository
```

3. Run “dpkg-scanpackages”

Switch to the repository directory and invoke the `dpkg-scanpackages` there:

```
cd /path/to/repository
dpkg-scanpackages -m . > Packages
```

This creates the `Packages` file holding the metadata for all packages in the newly created repository.

It is also possible to create a gzipped version of the `Packages` file - `Packages.gz`:

```
cd /path/to/repository
dpkg-scanpackages -m . | gzip > Packages.gz
```

4. Create a Debian configuration file for the repository

The configuration file (for example `myrepo.list`) should be placed under the `/etc/apt/sources.list.d/` directory and may look like:

```
deb [trusted=yes] file:/path/to/repository /
```

In the case when you’d like to share your repository over the network, e.g. to put it under `http://your-site.com`, the configuration file may look like:

```
deb [trusted=yes] http://your-site.com/your/repository/ /
```

or just use RpmDeb capabilities to create, host and distribute your packages worldwide without the neccessity to set up and maintain infrastructure.

Revision #1
Created 2023-03-31 21:11:35 UTC by willi
Updated 2024-11-02 15:34:58 UTC by willi