

easy filesystem hierarchy (efsh) v0.1.1

Nico Schottelius (nico-efsh-v0.1.1 at schottelius.org)

February 4, 2008

Contents

1	Preamble	4
1.1	Motivation	4
1.2	Disclaimer	4
1.3	Copying	4
1.4	Changes	4
1.5	Intended audience	4
1.6	Development	4
2	The easy filesystem hierarchy	4
2.1	Simplicity	4
2.2	Usability	4
2.3	Keep the root (/) clean	5
2.4	Do not mix different packages	5
3	The paths	5
3.1	/bin	5
3.2	/boot	5
3.3	/dev	5
3.4	/etc	5
3.4.1	/etc/.efsh	5
3.5	/home	6
3.5.1	/home/groups	6
3.5.2	/home/services	6
3.5.3	/home/users	6
3.6	/lib	6
3.7	/mnt	6
3.8	/proc	6
3.9	/root	6
3.10	/sbin	6
3.11	/tmp	6
3.12	/usr	7
3.13	/usr/bin	7
3.14	/usr/include	7
3.15	/usr/lib	7
3.16	/usr/packages	7
3.17	/usr/sbin	7
3.18	/usr/share	7
3.19	/var	7
3.19.1	/var/log	7

4	Operating specific paths	8
4.1	Linux	8
4.1.1	/sys	8
5	Obsoleted paths	8
5.1	/media	8
5.2	/opt	8
5.3	/usr/libexec	8
5.4	/usr/local	8
6	Misc	8
6.1	Real life examples	8
6.1.1	/etc/.efsh	8
6.1.2	/home/users/nico	8
6.1.3	/home/groups/finance	9
6.1.4	/home/services/ftp	9
6.1.5	/home/services/www/nico/org/schottelius/unix/www	9
6.1.6	/usr/packages/eboard-1.0.4	9
6.1.7	/usr/packages/wmii-hg	9
6.1.8	/usr/packages/wmii-hg-20071102.1	9
6.1.9	/usr/packages/xscreensaver-5.04/libexec/*	9
6.2	The future	9
6.3	Thanks go to	9
6.4	Changelog	10
6.4.1	v0.1.1: (4th of February 2008)	10
6.4.2	v0.1: Initial release (4th of February 2008)	10

1 Preamble

1.1 Motivation

Current Linux FHS is too complex. I want something short, that is easy to understand and maintain.

1.2 Disclaimer

This document is a pre-release.

1.3 Copying

Feel free to copy it unmodified.

1.4 Changes

Send changes to me via e-mail. Preferred in diff -u format to the \LaTeX source.

1.5 Intended audience

Sysadmins and Unix operating system creators.

1.6 Development

The source can be accessed via git:

- <http://unix.schottelius.org/cgi-bin/gitweb.cgi>
- <http://unix.schottelius.org/git/wichtige-dokumente.git>

2 The easy filesystem hierarchy

The main ideas of efsh, what it is about and how it looks like.

2.1 Simplicity

Today's systems tend to get more complex, resulting in unmaintainable status. Thus the base of your system must be as easy to understand as possible.

2.2 Usability

Admins are lazy and they need something they can remember intuitively.

2.3 Keep the root (/) clean

The root is seen in almost all situations and should contain only the essential directories. There should be no files in the root.

2.4 Do not mix different packages

Each software package should have its own area. So you can

- `rm -rf` it easily,
- see what files belong to it and
- install many different (or even same) versions in parallel.

3 The paths

Now it is time to define, what belongs to where.

3.1 /bin

Contains only basic binaries.

3.2 /boot

Contains system specific boot files and configuration.

3.3 /dev

Contains the device files.

3.4 /etc

Contains only essential configuration files.

3.4.1 /etc/.efsh

Contains the version of efsh the filesystem hierarchy is compliant to. As I really recommend to not pollute basic directories, I use a dot-file¹. This is necessary, because we need to place some kind of information file into the filesystem, that is early and easily available, so tools can detect the efsh version.

¹Files starting with a dot (".") are hidden files in Unix operating systems and not shown in a standard directory listing.

3.5 /home

Contains **all** data from users and servers. That means no data dispersion like in /usr/local/www and /usr/local/pgsql or /var/lib/www or ... **All** data belongs below **/home**. The main idea is to never make the sysadmin search for data, but to make it available in a single place.

3.5.1 /home/groups

Contains data from the groups. So people can easily work together.

3.5.2 /home/services

Contains data from the services.

3.5.3 /home/users

Contains data from the users. That means all home directories (with the exemption of the root users home directory).

3.6 /lib

Contains only basic libraries.

3.7 /mnt

Contains temporarily mounted filesystems. Permanently mounted filesystems belong to where they are needed (for instance /home/services/mail/queue).

3.8 /proc

Contains a virtual filesystem containing all processes.

3.9 /root

Home directory of the root user. Must be on / to prevent problems, if /home is not accessible.

3.10 /sbin

Contains only basic super user binaries. Permissions should be 0750, root:root.

3.11 /tmp

Can be used for temporary files.

3.12 /usr

Contains non basic software packages.

3.13 /usr/bin

Contains links to binaries under */usr/packages/*/bin*.

3.14 /usr/include

Same as in */usr/bin*, only headers are linked from */usr/packages/*/include*.

3.15 /usr/lib

Same as in */usr/bin*, only libraries are linked from */usr/packages/*/lib*.

3.16 /usr/packages

Contains the software packages in a subdirectory for each version:

- */usr/packages/gpm-2.0*
- */usr/packages/apache-2.2p12+hotfix42*
- */usr/packages/apache-2.2.1*
- ...

Each software package should contain its own *bin*, *include*, *lib*, *libexec*, *sbin*, *share*, ..., if needed.

3.17 /usr/sbin

Contains links to binaries under */usr/packages/*/sbin*. Same as in */usr/bin*, only super user binaries are linked from */usr/packages/*/sbin*.

3.18 /usr/share

Contains links to files under */usr/packages/*/share*.

3.19 /var

Contains variable data like logs.

3.19.1 /var/log

Contains the system logfiles.

4 Operating specific paths

There should be less possible. But as we have to carry on history, we have to support them until we find a better solution.

4.1 Linux

4.1.1 /sys

Contains kernel information.

5 Obsoleted paths

5.1 /media

Is senseless, as we have /mnt. Do not put unnecessary paths into the root directory.

5.2 /opt

Is replaced by /usr/packages.

5.3 /usr/libexec

Is not necessary, the binaries in libexec are only called by the same package (per definition) and are thus best placed into "/usr/packages/*name*/libexec".

5.4 /usr/local

There is no need to make a mess in /usr/local with different software packages, if each package can have its own mess in /usr/packages/*name*.

6 Misc

To make it easier to understand how it looks like.

6.1 Real life examples

Efsh is not only theory, but already in use.

6.1.1 /etc/efsh

Contains "0.1.1" (without the quotes, including a new line).

6.1.2 /home/users/nico

This is my home directory on all servers and hosts I manage.

6.1.3 `/home/groups/finance`

Contains documents from the finance department. The directory has ACLs² applied, so every new document is read and writable by all members of the finance group.

6.1.4 `/home/services/ftp`

Contains the ftp root directory.

6.1.5 `/home/services/www/nico/org/schottelius/unix/www`

Contains the www data of *unix.schottelius.org*, which is owned by *nico*. The access and error log is placed below */home/services/www/nico/org/schottelius/unix/logs*, which is **not** owned by *nico*, but by the *www* user.

6.1.6 `/usr/packages/eboard-1.0.4`

Eboard in version 1.0.4.

6.1.7 `/usr/packages/wmii-hg`

Wmii installed from mercurial (hg).

6.1.8 `/usr/packages/wmii-hg-20071102.1`

Wmii installed from mercurial (hg), but with the version appended, so I can have more than one version from mercurial installed.

6.1.9 `/usr/packages/xscreensaver-5.04/libexec/*`

Screensavers from the xscreensaver package.

6.2 The future

Efsh is in a quite early development state and is missing non-Linux operating specific details. Though it will not grow much anymore, as it is intended to be small.

6.3 Thanks go to ...

In chronological order

- Dominik Meister
 - For correcting typos

²Access control lists

- Tonnerre Lombard
 - For critics

6.4 Changelog

The changelog is placed at the end of the document, so interested readers **can** find it, but new readers are **not forced** to read it.

6.4.1 v0.1.1: (4th of February 2008)

- Added changelog section
- Added examples section
- Fixed some typos
- Added new paths:
 - /etc/.efsh
 - /usr/libexec
 - /usr/local
 - /var/log
 - /tmp

6.4.2 v0.1: Initial release (4th of February 2008)

First time written down in an easy-to-change (L^AT_EX) and easy-to-view (pdf) format.