Haiku Distro Guidelines

A debugging

by Nicholas Blachford
But first...
It’s been a long time in the making...

(OpenBeOS demoed at BeGeistert 010 )
Congratulations on Haiku Alpha 1!
Why do the guidelines exist?

- To create a single standard OS
- Ensures compatibility
- Avoids “Distro hell”
So, what seems to be the problem?

The Guidelines are:

• Self contradictory
• Vague
• In some cases, might be impossible to follow
• Based on a false assumption
• Nothing is explained
• Liable to lead to a great deal of confusion
• More likely to lead to disto hell!
Self contradictory

- “Haiku distributions must comply to the following set of guidelines”
- The MIT license defines the rules. The guidelines are voluntary
- Fix: change “must” to “should”
- Don’t order people to follow the guidelines
  - They will ignore you!
- Encourage people to follow the guidelines
Self contradictory 2

- “The trademark "Haiku" may not be used in your distribution's name.”
- This is covered by trademark law
- You lose copyrights if you do not enforce them
- Fix: change “may” to “must”
- Trademark rules should be separate in the guidelines

- I’ll come back to this later...
Self contradictory 3

- Use the "common" folder for customizations whenever possible.
- Is “common” a good name for a folder that holds things that are not common?

- How about:
  - /custom/my_distro_name/
Vague

• “You must not change or extend the API of the Haiku system libraries in any way”

• Why not?
• What are the “Haiku system libraries”?
• Can I add APIs in other libraries?
• Can I add other libraries?
Vague 2

• You should not change, move, or rename any of the files and folders that are part of the base distribution's "system" folder without good reason.

• What is a good reason?
• What is a bad reason?
• What is the base distribution?
Impossible to follow?

- "Use the same GCC version as the official distribution (X86 GCC 2.95.3 Hybrid with GCC 4.3.3 alternative GCC libraries)"

- Making an ARM port compliant is a bit difficult if you have to compile for x86!

- What if you are porting to a CPU arch that does not have that version of gcc?

- What if BeOS compatibility is not a goal of your distro?
Nothing is explained

- Why do these guidelines exist?
- What is their purpose?
- Why should I follow them?
- What do I get if I do?
- What do I lose if I don’t?
The *real* problems

I’ve just been pedantic so far...
What is Distro Hell?

• Linux has lots of distros
• ...but many are incompatible

• The guidelines were written to avoid this
Why does distro Hell exist?

Distro Hell is an historical problem with Linux:

• Technically, Linux is only a kernel
• There is no user land
• When Linux started, distro makers had to define their own user land.
• They all did it their own thing
• They added complex package management
• The problem exists to this day...
The false assumption

- “Multiple Haiku distros automatically lead to distro Hell”

- Distro Hell is unique to Linux, it does not affect any other OS
Will distro Hell affect Haiku?

- Haiku is starting with a well defined user land

- Unless you need to make big changes to the system, distro Hell for Haiku should simply not exist
But here is a problem...

• Q - How do you create an *interesting* distro if you cannot be different?

• A - You can’t. The guidelines do not allow it

• It is going to happen anyway
Haiku is part of the open source world

- A world where nobody can agree on anything
- “Lots of communities that all hate each other”
- Projects change and fork for the dumbest of reasons
- but in many cases this has lead to better things...
Open source likes to fork

- X.org - forked because of license dispute
- OpenSSH - started because of license dispute
- PF - started because of license dispute
- Gnome - started because of KDE license dispute
- gcc - started as an experiment
- Firefox - started as a side project
Even OSs like to Fork

OpenBSD

• A personality clash lead to the creation of the definitive secure OS

DragonFly BSD

• A technical disagreement lead to a system that is adding clustering into the kernel. Eventually all computers will become clusters
Change leads to good things
Mass confusion

• Not allowing the Haiku name to be used by distros is going to cause a lot of confusion
A tale of 2 distros

• Compliant distro “myOS” is fully compliant and compatible
• ...but does not sound like Haiku
• If you want to solve problems you have to ask on a website about an OS that sounds completely different from yours!
A tale of 2 distros

- Non-compliant distro “HighKoo” is non-compliant and incompatible, but does sound like Haiku.

- Any advice from a Haiku website is likely to be confusing at best.

- HighKoo isn’t trademarked so there’s nothing to stop someone using it.
A tale of 2 distros 3

- Remember, guidelines are voluntary
- Only one is compliant
- Both are “Haiku distros”
- The name is no help in telling you which is which
If they change name

- The non-compliant “HighKoo” becomes “AnotherOS”

- But, even if they don’t directly use the name, it will still be a “Haiku distro”, whether it follows the guidelines or not

- The trademark guidelines give you no way to tell what is compliant and what is not
What's going on?

- Linux has a generic family name
- Ubuntu, RedHat, Suse are the brands
- BSD has a generic family name
- FreeBSD, OpenBSD, NetBSD are the brands
- Haiku does not have a family name to distinguish individual brands
What’s going wrong? 2

- This is the open source world
- There are many different ways of doing things
- The open source world does all of them
- Changes will happen
- The trademark rules as they stand will only make things worse
The ultimate problem

• The guidelines do not allow for change

• They are designed to preserve a system, but ignore anyone who changes it

• Change is going to happen if you like it or not

• If you do not allow change, you cannot control it

• If no direction is given, they will take their own
The current guidelines lead directly to distro Hell!
You don’t have to go there
The important thing

• Lots of compatible distros is not a problem

• Lots of incompatible distros is
The important thing: Compatibility

- Source compatibility - works across all platforms, all compiler versions
- Binary - more user friendly but very limiting
Accept change

- Having a single OS is a good idea
- But, people will want to change it
- The guidelines should accept this
Accept change 2

- Define change so distros can remain compatible with Haiku and other Haiku distros
- Define what can change
- Define what should not change
Enforcing the guidelines

The Java EE model:

- Java EE defines a base system
- If you are compliant with the base system you can say you are a Java EE system
- You can add whatever you want beyond that
Enforcing the guidelines 2

• Define Haiku as a “base system”
• Define a very specific set of rules that define compatibility with the base system
• Enforce this with a test suite
• Do not define anything that might be due to personal preference (look and feel, spatial browsing etc.)
• Allow distro developers as much freedom as possible
Enforcing the guidelines 3

- The Haiku name is by far the best weapon to use in enforcing guidelines
- Define a name for the family: Haiku seems obvious
- Define a name for the base system: e.g. Base Haiku?
- Allow distros to use the family name if they can prove they are compatible with the base system
- Define a naming scheme:
  - OpenHaiku, FreeHaiku, NetHaiku
Enforcing the guidelines 4

• Even with well defined guidelines there will still be some who cannot* act within them

• Even here it might be possible to contain potential damage:
  • Define the specific sub-systems within Haiku
  • If one sub-system is changed it should not affect the others

*There might be a very good reason for this
Conclusion

• Guidelines are a good thing
• They need to define:
  • Why they exist
  • Binary and source compatibility
  • The base system
  • What can change
  • What should not change
  • How to minimise unsafe changes
  • Define a family and a brand
  • Rules for the use of trademarks
The end