

The Haiku Package Manager

Richard Zak
richard.j.zak@gmail.com

10 November 2021



1 Introduction

2 Packages!

- Package Manager
- Package Metadata
- Considerations when Porting

3 Demo

4 Closing



whoami

- My name is Richard Zak
- Haiku enthusiast, software developer, cybersecurity researcher
- I'm not a member of the Haiku team
- Tinkering with computers since the mid-90's
- <https://rjzak.github.io/>
- richard.j.zak@gmail.com



Haiku: History

From Haiku's website:

Haiku is an open-source operating system that specifically targets personal computing. Inspired by the BeOS, Haiku is fast, simple to use, easy to learn and yet very powerful.



BeOS image from ArsTechnica.com.



What is Haiku?

- Single-user, Unix-inspired, desktop operating system
 - ▶ Executables and libraries are ELF's
 - ▶ Bash is the default shell for the Terminal
 - ▶ SSH is included, client & server
 - ▶ Many programs work on Haiku with a simple recompile, simple build changes, simple header changes.
 - ▶ Most programs are compiled with Make, CMake, gcc. LLVM is available too.
 - ▶ It's not Linux, BSD, etc. Haiku has it's own unique kernel.
- Inspired by BeOS, which was inspired by the Classic Mac OS
- Exists solely because of it's developer & user community, volunteer efforts, and donations. There's no corporate sponsorship.
- Now works on RISC-V. Work is in progress for ARM64 support, plus others.
- ... and it's 20 years old! The project was formed in 2001, with it's first release in 2002.



Haiku Packages

The usual features:

- Separate packages for the application, development, source
- Retrieval of packages, including dependencies, from a repository
- Architecture-aware

The screenshot shows the Haiku Package Manager window. At the top, there's a search bar with 'java' entered. Below it, a table lists search results. The table has columns for Name, Rating, Description, Size, Status, and Version. Three packages are visible: 'openjdk13_sources', 'openjdk14', and 'openjdk14_default'. The 'openjdk14' package is selected. Below the table, there's a detailed view for 'openjdk14', including a star rating, version number (14.0.2.12-1), and an 'Install' button. The description for 'openjdk14' reads: 'An open-source implementation of the Java Platform, SE'. Below the description, there's a paragraph about the GNU General Public License (GNU GPL) and a link to the official Java SE 8 reference implementation.

Name	Rating	Description	Size	Status	Version
openjdk13_sources	★★★★★	JDK source files, demos and examples	55.30 MiB	Available	13.0.2.8-4
openjdk14	★★★★★	An open-source implementation of the Java Platform, SE	75.30 MiB	Available	14.0.2.12-1
openjdk14_default	★★★★★	An open-source implementation of the Java Platform, SE	1.63 KiB	Available	14.0.2.12-1

openjdk14 ★★★★★ n/a 14.0.2.12-1 Install

An open-source implementation of the Java Platform, SE

OpenJDK (Open Java Development Kit) is a free and open source implementation of the Java Platform, Standard Edition (Java SE). It is the result of an effort Sun Microsystems began in 2006.

The implementation is licensed under the GNU General Public License (GNU GPL) with a linking exception. Were it not for the GPL linking exception, components that linked to the Java class library would be subject to the terms of the GPL license. OpenJDK is the official Java SE 8 reference implementation.

<https://openjdk.java.net/>

Haiku Packages

The usual features:

- Separate packages for the application, development, source
- Retrieval of packages, including dependencies, from a repository
- Architecture-aware

Some unique features

- Package contents are mounted as read-only into the file system
- Packages can indicate which libraries and/or commands are provided, allowing for easier dependency management
 - ▶ Packages can have a dependency specified as `lib:jpeg` vs. `libjpeg-2.0.3`. Because it is `libjpeg` or `libjpeg-turbo8`, for example?
- The file system knows which package is the origin for a given file.
- The boot loader knows about the package manager.



Packaged Files & Directories are Read-Only

Since the package contents are mounted from the package itself on to the file system, the files and directories are read-only.

- Made possible by separating the package file into equal-sized chunks and compressing individually, enabling cheap random access into the compressed file.
- This benefits the user, since it's guaranteed that the file can't be modified, either accidentally or intentionally.
- This benefits the developer, since it's guaranteed that the program and it's supporting files will be present in the known relative paths.
- The user can side-step this, if desired, to install software in `/boot/system/non-packaged/`
- Since the installed packages are on disk, making them redistributable (if offline, for example) is possible.



The Boot Manager is Package-Aware

Welcome to the
Haiku Boot Loader

Copyright 2004-2020 Haiku, Inc.

Select Haiku version

Latest state

```
2021-06-26 11:24:46
2021-06-26 11:24:25
2021-06-26 11:23:38
2021-06-26 11:23:29
2021-06-26 11:23:21
2021-06-26 11:23:12
2021-06-26 11:22:58
2021-06-26 11:22:50
2021-06-26 11:22:42
2021-06-26 11:22:23
2021-06-26 00:05:51
2021-06-24 23:17:01
```



The Boot Manager is Package-Aware

- Prior states allowing going back to how the system was previously
 - ▶ Including Haiku itself, since it's also in a package.
- Immensely helpful for operating system development, debugging
- Prevents the user from being completely locked out of their system



Metadata in Packages

Package information:

- Basics: name, description, software license
- Provides: indicates if a package provides a library, `lib:foo` or command, `cmd:bar`
- Other packages, and the command line, can specify dependencies on by referring to the the library name or command without having to specify which package exactly.
- Example: `pkgman install cmd:foo`



Considerations when Porting

Generally, from a packaging perspective, porting to Haiku isn't too difficult. But there are some things to be considered:

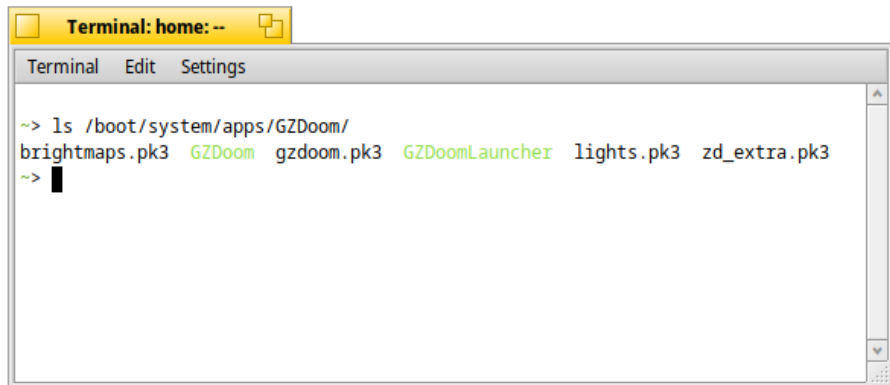
- A lot of the same files exist on Haiku, but in different locations.
 - ▶ `/etc/resolv.conf` → `/boot/system/settings/network/resolv.conf`
 - ▶ Configuration dot files in the home directory *should* now be in `/boot/home/config/settings/` (convention)
- Directories from a package are read-only, so, for example, using Python's `pip` becomes problematic, as `/boot/system/lib/python3.7/` and its subdirectories are read-only. Solution:
 - ▶ Package the desired module into a Haiku package, or
 - ▶ install the package in `/boot/system/non-packaged/lib/python3.7/`, or
 - ▶ create a virtual environment in the home directory, `/boot/home`.



Demo!

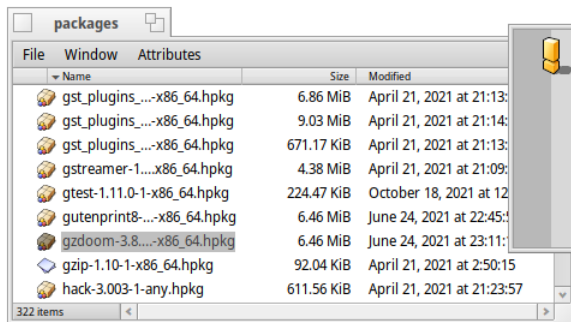
There's another neat feature of Haiku's packaging system. . .

The Filesystem is Package-Aware



```
Terminal: home: --  
Terminal Edit Settings  
~> ls /boot/system/apps/GZDoom/  
brightmaps.pk3  GZDoom  gzdoom.pk3  GZDoomLauncher  lights.pk3  zd_extra.pk3  
~> █
```

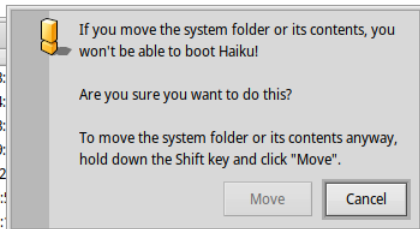
The Filesystem is Package-Aware



A screenshot of a file manager window titled "packages". The window displays a list of files with columns for Name, Size, and Modified. The files listed are:

Name	Size	Modified
gst_plugins_...-x86_64.hpkg	6.86 MiB	April 21, 2021 at 21:13:
gst_plugins_...-x86_64.hpkg	9.03 MiB	April 21, 2021 at 21:14:
gst_plugins_...-x86_64.hpkg	671.17 KiB	April 21, 2021 at 21:13:
gststreamer-1...-x86_64.hpkg	4.38 MiB	April 21, 2021 at 21:09:
gtest-1.11.0-1-x86_64.hpkg	224.47 KiB	October 18, 2021 at 12:
gutenprint8-...-x86_64.hpkg	6.46 MiB	June 24, 2021 at 22:45:
gzdoom-3.8...-x86_64.hpkg	6.46 MiB	June 24, 2021 at 23:11:
gzip-1.10-1-x86_64.hpkg	92.04 KiB	April 21, 2021 at 2:50:15
hack-3.003-1-any.hpkg	611.56 KiB	April 21, 2021 at 21:23:57

The status bar at the bottom of the window indicates "322 items".



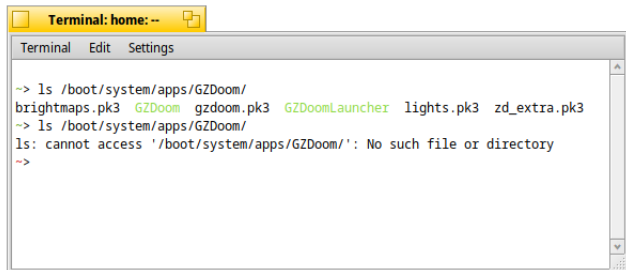
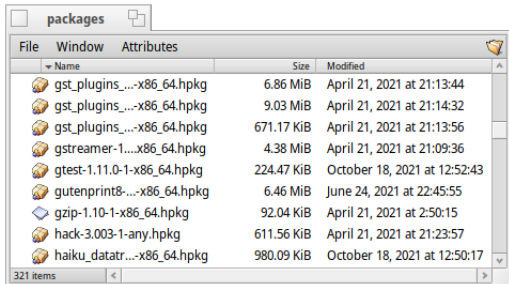
If you move the system folder or its contents, you won't be able to boot Haiku!

Are you sure you want to do this?

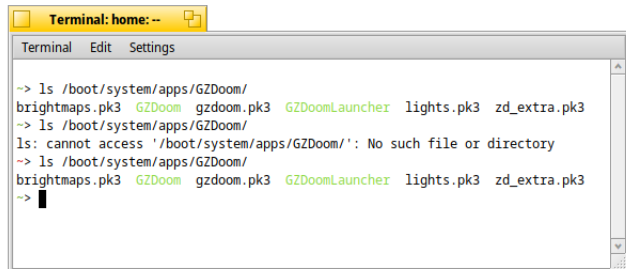
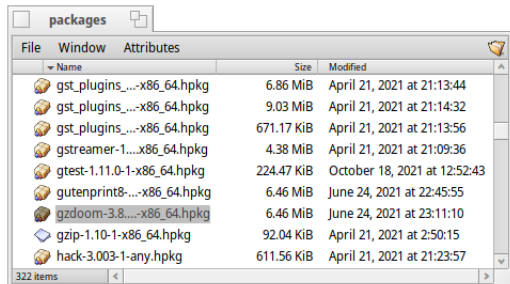
To move the system folder or its contents anyway, hold down the Shift key and click "Move".

Move Cancel

The Filesystem is Package-Aware



The Filesystem is Package-Aware

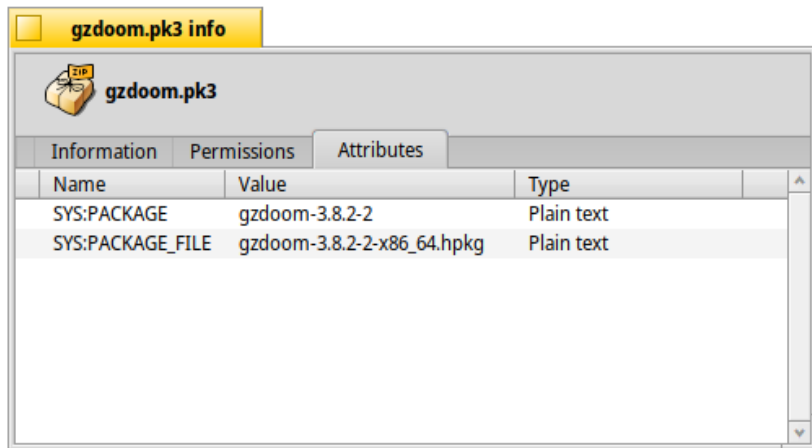


The Filesystem is Package-Aware

- Removing a package from the packages directory, `/boot/system/packages/`, uninstalls the package.
 - ▶ The system warns you that is is probably a bad idea.
 - ▶ It can be done anyway by holding the shift key.
- Putting the package back into the packages directory re-installs the package.



The Filesystem is Package-Aware




The screenshot shows a file manager window with a yellow title bar that reads "gzdoom.pk3 info". Below the title bar is a grey header area with a ZIP icon and the text "gzdoom.pk3". Underneath is a tabbed interface with three tabs: "Information", "Permissions", and "Attributes". The "Attributes" tab is active and displays a table with the following data:

Name	Value	Type
SYS:PACKAGE	gzdoom-3.8.2-2	Plain text
SYS:PACKAGE_FILE	gzdoom-3.8.2-2-x86_64.hpkg	Plain text


The BeFS attribute `SYS:PACKAGE_FILE` shows the package which is the source for a given file.


Questions?

Learn more about Haiku: <https://www.haiku-os.org/>

 <https://github.com/haikuports/haikuports/>

IRC: #haiku on OFTC

 <https://discuss.haiku-os.org>

 <https://twitter.com/haikuOS>

 <https://discord.gg/8KsjHbW>



Creating Packages

- Packages are created with a simple text file which contain metadata, typically in the root directory of the files to be packages, named “.PackageInfo”
- Required fields:
 - ▶ name
 - ▶ version
 - ▶ architecture – Specific architecture, list of architectures, or “any”
 - ▶ summary
 - ▶ description
 - ▶ packager – Person who packaged the software
 - ▶ licenses – Must be a license name known to the packager
 - ▶ provides – List of software names, executables, and/or libraries and version number
 - ▶ requires – List of required packages and/or libraries, optionally with version requirements



Creating Packages

- Directory contains the directories & files to be packages
 - ▶ `bin/` in the project will be mounted as `/boot/system/bin/package_name/`
 - ▶ `lib/` in the project will be mounted as `/boot/system/lib/package_name/`
- Command: `package create -C /path/to/dir/ -b Output.hpkg`
- Example: https://github.com/rjzak/ghidra/blob/master_haiku64/.PackageInfo



HaikuPorts

- Haiku has a vast collection of ported software, called HaikuPorts.
 - ▶ <https://github.com/haikuports/haikuports>
- HaikuPorts contains recipes to build software, and patches (if needed) to enable or improve Haiku support. These are built automatically and added to the repository.
- HaikuPorter helps build applications from HaikuPorts, and is used by the main package repository but can be run locally.
 - ▶ <https://github.com/haikuports/haikuporter>
 - ▶ <https://github.com/haikuports/haikuports.cross> for cross compiling for different architectures, with RISC-V getting a lot of attention recently

