habitat

habitat.sh | habitat.sh/docs

HABITAT CHEAT SHEET

basics

`plan.sh`	A shell script you use to define the build lifecycle of your application code.
`hooks/`	A directory you store the run lifecycle of your application artifact.
`config/`	A directory to store any configuration files your application needs to run. Config files can be templated using the handlebars templating language.
`default.toml`	A TOML file containing default settings for any config settings that are templated in **config/**

plans

Basic Setting: Every **plan.sh** starts with some basic settings.

`pkg_name` Required. Sets the name of the package.

```
pkg_name=zlib
```

`pkg_origin` Required. Sets the origin of the package.

```
pkg_origin=myorigin
```

`pkg_version` Required. Sets the version of the package.

```
pkg_version=1.2.8
```

`pkg_maintainer` Optional. The name and email address of the package maintainer.

```
pkg_maintainer="Your Name <someone@example.com>"
```

`pkg_license` Optional. An array of valid software licenses that relate to this package.

```
pkg_license=('Apache-2.0')
```

`pkg_source` Required. A URL that specifies where to download the source from.

```
pkg_source=http://somehost.tld/$pkg_name/${pkg_
version}/${pkg_name}-${pkg_version}.tar.gz
```

`pkg_deps` Optional. An array of package dependencies needed at runtime.

```
pkg_deps=(core/glibc core/pcre core/openssl core/zlib)
```

`pkg_build_deps` Optional. An array of the package dependencies needed only at build time.

```
pkg_build_deps=(core/gcc core/linux-headers)
```

call backs

The Build Lifecycle: Call Backs are functions you can use to define the different steps of how your application code is taken from source code to an artifact that can be deployed. Override the default Call Back actions by defining one of these functions in your **plan.sh**.

`do_begin()`	Default Implementation: No - Useful for anything that needs to happen before the plan starts.
`do_download()`	Default Implementation: Yes - Download the source package.
`do_verify()`	Default Implementation: Yes - Verify the package checksum
`do_clean()`	Default Implementation: Yes - Cleans the build environment
`do_unpack()`	Default Implementation: Yes - Extract the source archive. Supported formats are .tar, .tar.bz2, .tar.gz, .tar.xz, .rar, .zip, .Z, & .7z.
`do_prepare()`	Default Implementation: No - Set any variables or run anything before the software is built
`do_build()`	Default Implementation: Yes - Runs **make** by default
`do_check()`	Default Implementation: Yes - Does nothing by default. Typically for running **make test**
`do_install()`	Default Implementation: Yes - Runs **make install** by default
`do_strip()`	Default Implementation: Yes - Strips binaries by default
`do_end()`	Default Implementation: No - Useful for post build clean up

hooks

The Run Lifecycle

`init`	File location: **<plan>/hooks/init** This hook is only run when the Habitat supervisor starts the first time.
`run`	File location: **<plan>/hooks/run**
`file_updated`	File location: **<plan>/hooks/file_updated**
`reconfigure`	File location: **<plan>/hooks/reconfigure** This hook is run when service configuration information has changed through a set of Habitat services that are peers with each other.
`health_check`	File location: **<plan>/hooks/health_check** This hook is run when the Habitat HTTP API receives a request at **/health**. The **health_check** script must return a valid exit code from the list below.

0 - ok, **1** - warning, **2** - critical, **3** - unknown, **Other** - failed health check with additionaloutput taken from **health_check** stdout

config files

Store any config files for your application in **config/**. Template them using the Handlebars templating language. Default values should be placed in the **default.toml**.

Handlebars Basics Learn more at handlebarsjs.com

```
{
    "message": "{{cfg.message}}",
    "port": "{{cfg.port}}"
}
```

`default.toml`

```
# Default values that can be updated.
# Message of the Day
message = "Hello, World!"
# The port number that is listening for requests.
port = 8080
```