

---

# **ameh Documentation**

***Release 0.0.1***

**Eric Pierce**

July 17, 2012



# CONTENTS

<b>1</b>	<b>Motivation</b>	<b>3</b>
<b>2</b>	<b>Configuration</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Permissions . . . . .	7
3.2	init . . . . .	7
3.3	config . . . . .	8
<b>4</b>	<b>Development</b>	<b>9</b>
<b>5</b>	<b>Ideas</b>	<b>11</b>
<b>6</b>	<b>Indices and tables</b>	<b>13</b>



Tools for managing a set of [Atlassian](#) product installations.

[Source code on github.](#)

Contents:



# MOTIVATION

This project was born out of recurring annoyances with Atlassian products, specifically:

- Lack of `init.d` service scripts to allow automatic startup at boot time
- Non-standard and scattered configuration files and formats
- Inconsistently-named and -located executables for starting and stopping servers
- Too many user accounts created (jira1, jira2 etc.) when installing

The name “ameh” came from “atlas meh”, which you could see as a synonym for “atlas shrug” if you’re into that sort of thing. If you like acronyms, you could think of it as “Atlassian MaintenanceE Helper”. The goal of this project is to make me (and possibly others) have a “meh, no problem” attitude towards Atlassian product maintenance, instead of the “please fire me and put me out of my misery” attitude that I have now.





# CONFIGURATION

ameh stores its configuration in a file like this:

```
[jira]
install = /opt/atlassian/jira
home = /var/atlassian/application-data/jira-home
classes = %(install)s/atlassian-jira/WEB-INF/classes
crowd = %(classes)s/crowd.properties
db = %(home)s/dbconfig.xml

[confluence]
install = /opt/atlassian/confluence
home = /var/atlassian/application-data/confluence-home
db = %(home)s/confluence.cfg.xml

# Any apps you don't use will have their sections commented out
#[crowd]
#[fisheye]
```

New configuration settings can be added using the ameh command-line, or by directly editing the configuration file (which will typically live in `/etc/ameh.ini`).



# USAGE

The main way to use ameh is through the command-line script ameh, which includes several subcommands for doing various tasks. Run ameh with no arguments to get help, or ameh <command> with no arguments to get help on a particular command.

## 3.1 Permissions

All ameh commands are designed to be run as a normal user, without requiring root access. It's recommended that you use sudo only when you actually need elevated permissions for something.

## 3.2 init

Atlassian products tend not to include /etc/init.d scripts for starting, stopping, or checking the status of your applications. The init command aims to remedy this by providing an init-script generator:

```
$ ameh init jira

#!/bin/sh -e
# /etc/init.d/jira

# [runlevels] [start order] [stop order]
# chkconfig: 2345 80 20
# description: jira

APP="jira"
USER="jira"
START="/opt/atlassian/jira/bin/startup.sh"
STOP="/opt/atlassian/jira/bin/shutdown.sh"

case "$1" in
    start)
        echo "Starting $APP"
        /bin/su -m $USER -c "$START &> /dev/null"
        ;;
    stop)
        echo "Stopping $APP"
        /bin/su -m $USER -c "$STOP &> /dev/null"
        ;;
    restart)
        $0 stop
```

```
        sleep 5
        $0 start
        ;;
*)
    echo "Usage: $0 {start|restart|stop}"
    exit 1
    ;;
esac
```

This command simply generates an init-script based on the settings you have configured in `/etc/ameh.ini`, and prints it on standard output. This way you, being a responsible sysadmin, can review everything it does before installing it into your `/etc/init.d` folder. To do that, you can pipe into a `tee` command with elevated permissions:

```
$ ameh init jira | sudo tee /etc/init.d/jira
```

You may need to customize the resulting script for your particular environment, but this will at least give you a starting point.

### 3.3 config

Jira, Confluence, and other Atlassian products have configuration files scattered all over the place. It can be tedious trying to remember where each one is. The `config` command can help you keep track of them.

When you run `ameh config <app>`, all of the properties defined in the `[app]` section of your `/etc/ameh.ini` file are printed. For example, you might get something like:

```
$ ameh config jira
jira install: /opt/atlassian/jira
jira start: /opt/atlassian/jira/bin/startup.sh
jira stop: /opt/atlassian/jira/bin/shutdown.sh
jira home: /var/atlassian/application-data/jira-home
jira classes: /opt/atlassian/jira/atlassian-jira/WEB-INF/classes
jira db: /var/atlassian/application-data/jira-home/dbconfig.xml
```

Some of these are configuration files, and others are directories or shell scripts. If it's configured, it's printed here. Let's say you want to modify your database configuration, and you don't remember which file that's in:

```
$ ameh config jira db
/var/atlassian/application-data/jira-home/dbconfig.xml
```

To edit this file, you can simply surround that command in backticks:

```
$ vim `ameh config jira db`
```

Or, if you need elevated permissions:

```
$ sudo vim `ameh config jira db`
```

# DEVELOPMENT

It's a good idea to run ameh in a virtualenv, especially during development. The short version:

```
$ pip install virtualenvwrapper
$ mkvirtualenv ameh
```

Python packages needed for development are listed in `dev-req.txt`; you can install them with pip like so:

```
$ pip install -r dev-req.txt
```

Due to the preponderance of really old Python versions in enterprise environments like CentOS, the code in ameh is designed to be compatible with Python 2.4. If you contribute new code to ameh, please avoid using features from newer versions of Python.

Unit tests are in the `tests` directory. You can run them with `py.test` like this:

```
$ py.test
```



# IDEAS

Some tools that it might be worthwhile to include:

- init.d scripts, autogenerated based on configured install/home directories
- Backup scripts
- Symlinking scripts, making configuration files and logs easier to manage by having them all in one place (preferably somewhere standard like `/etc` and `/var/log`)
- User/group maintainer to prevent multiple jira1, jira2 users from being created, and to run each service as a dedicated user

Will probably need:

- A single configuration file to define which Atlassian products you have installed, where their install/home directories are, what database backend you're using, etc.
- A nice wrapper executable that can be installed system-wide





# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*