

Boot Documentation Library

Boot User Guide

Gregg Reynolds

Boot User Guide (DRAFT)

abstract

draft intro...

Table of Contents

Part 1: Using Boot	1
Boot Tasks	2
Middleware Concepts	2
Anatomy of a Boot Task	2
Boot Pipelines	3
Pipeline Concepts	3
Boot Filesets	4
Fileset Roles	4
Boot Workspaces	5
Workspace Concepts	5
Part 2: Reference Manual	6
Boot Standard Task Library	7
add-repo	7
aot	7
install	7
help	8
jar	8
javac	9
pom	9
push	9
repl	10
show	10
sift	11
target	12
uber	12
wait	13
war	13
watch	14
web	14
zip	14
Boot API	16
boot.core	16
input-files	16
tmp-dir	16
boot.pod	16
Appendix	18

Guide Abstract

The content of this UG is drawn from the [Boot Wiki](#) and the [Boot Github repo](#).

Summary

Notice

This information was developed for products and services offered in the U.S.A.

This product is meant for educational purposes only. Some of the trademarks mentioned in this product appear for identification purposes only. Not responsible for direct, indirect, incidental or consequential damages resulting from any defect, error or failure to perform. Any resemblance to real persons, living or dead is purely coincidental. Void where prohibited. Some assembly required. Batteries not included. Use only as directed. Do not use while operating a motor vehicle or heavy equipment. Do not fold, spindle or mutilate. Do not stamp. No user-serviceable parts inside. Subject to change without notice. Drop in any mailbox. No postage necessary if mailed in the United States. Postage will be paid by addressee. Post office will not deliver without postage. Some equipment shown is optional. Objects in mirror may be closer than they appear. Not recommended for children. Your mileage may vary.

No other warranty expressed or implied. This supersedes all previous notices.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Retro Tools, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the Retro Tools in the United States, other countries, or both:

RetroWrench®

The following terms are trademarks of other companies:

Red, Orange, Yellow, Green, Blue, Indigo, and Violet are registered trademarks of Rainbow Corporation and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed finibus rutrum pharetra. Etiam porttitor purus non felis semper, vel lobortis sem ullamcorper. Donec ut interdum turpis, non placerat lacus. Quisque placerat lacus id magna rhoncus, nec lacinia massa blandit. Pellentesque faucibus, dolor vitae accumsan pretium, arcu mauris eleifend felis, a iaculis justo nisl vel quam. Fusce laoreet turpis et finibus molestie. Suspendisse maximus scelerisque dui, vel vestibulum libero porttitor id. In et libero erat. Integer et dolor eget tellus dictum fermentum. Nunc velit elit, eleifend et placerat non, convallis eget mauris. Fusce congue ipsum ac commodo tincidunt. Mauris varius vulputate ante sit amet consequat. Cras et finibus est, fringilla vestibulum turpis. Quisque consectetur felis at nibh vulputate, id bibendum sapien venenatis. Mauris dapibus accumsan ornare.

About this Document

This document ...

Revision: 0.1

Acknowledgements

acks here...

Part 1: Using Boot

This is the overview of Part I ...

Fusce porta leo sem, non luctus lectus fermentum et. Nunc lacus mi, ultricies et ex sit amet, porttitor convallis ligula. Praesent convallis nibh id lectus pellentesque, et rutrum nibh molestie. Nunc ac convallis lectus. Aliquam eu sem eget est mollis iaculis quis eu mauris. Vestibulum iaculis turpis a urna sagittis, vitae pellentesque augue venenatis. Fusce ut efficitur arcu. Mauris volutpat velit quis purus consequat, sit amet molestie mauris aliquam. Curabitur dolor eros, congue eget erat sed, suscipit auctor erat. Cras convallis ex in sem placerat ornare non at felis.

Boot Tasks

A boot *task* is ...

Related information

- [Tasks \(Wiki page\)](#)

Middleware Concepts

The concept of *middleware* is central to boot's concept of a task.

Boot borrows the notion of middleware from [Ring](#).

Important

Boot's notion of "middleware" is a little bit peculiar, and should not be confused with other [common uses of the term](#).

Anatomy of a Boot Task

A boot task has the following structure:

Boot Pipelines

A boot *pipeline* is ...

Pipeline Concepts

Pipeline processing...

Pipelines are actually two-way, or more accurately, there are two pipelines, connected at the extremities, running in opposite directions. So it's really a circuit rather than a pipeline. Electrical flow is probably a better metaphor than water flow.

Boot Filesets

A boot *fileset* is ...

Related information

- [Filesets \(wiki page\)](#)

Fileset Roles

There are three roles ...

Boot Workspaces

A boot *workspace* is ...

An important principle of the boot build process is that tasks do not refer to named places in the filesystem. Tasks may only create files in managed temp directories provided by boot. These temp directories are:

- Anonymous – tasks do not specify the location of the temp dir.
- Local – tasks do not pass references to temp dirs to other tasks.
- Managed – temp dirs are cleaned up by boot as necessary.

In order to communicate files in these temp directories to the rest of the build process they must be added to the fileset object, described below.

Related information

- [Tasks \(Wiki page\)](#)

Workspace Concepts

"Workspace" is a metaphor for ...

Part 2: Reference Manual

This is a reference manual for Boot.

Fusce porta leo sem, non luctus lectus fermentum et. Nunc lacus mi, ultricies et ex sit amet, porttitor convallis ligula. Praesent convallis nibh id lectus pellentesque, et rutrum nibh molestie. Nunc ac convallis lectus. Aliquam eu sem eget est mollis iaculis quis eu mauris. Vestibulum iaculis turpis a urna sagittis, vitae pellentesque augue venenatis. Fusce ut efficitur arcu. Mauris volutpat velit quis purus consequat, sit amet molestie mauris aliquam. Curabitur dolor eros, congue eget erat sed, suscipit auctor erat. Cras convallis ex in sem placerat ornare non at felis.

Boot Standard Task Library

Boot comes with a set of built-in tasks.

add-repo

```
$ boot add-repo --arg ...
```

```
(add-repo :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [add-repo wiki page](#).

aot

```
$ boot aot --arg ...
```

```
(aot :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [aot wiki page](#).

install

```
$ boot install --arg ...
```

```
(install :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

• [aot wiki page](#).

help

```
$ boot help --arg ...
```

```
(help :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

• [help wiki page](#).

jar

```
$ boot jar --arg ...
```

```
(jar :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [jar wiki page](#).

javac

```
$ boot javac --arg ...
```

```
(javac :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [javac wiki page](#).

pom

```
$ boot pom --arg ...
```

```
(pom :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [pom wiki page](#).

push

```
$ boot push --arg ...
```

```
(push :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [push wiki page](#).

repl

```
$ boot repl --arg ...
```

```
(repl :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [repl wiki page](#).

show

```
$ boot show --arg ...
```

```
(show :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [show wiki page](#).

sift

```
$ boot sift --arg ...
```

```
(sift :arg ...)
```

DESCRIPTION

Transform the filesset, matching paths against regexes.

Arguments:

:add-asset

-A

`{str}` The set of directory paths to add to assets.

:add-jar

-j

`{sym regex}` The map of jar to path regex of entries in jar to unpack.

:add-meta

-M

`{regex kw}` The map of path regex to meta key to add.

:add-resource

-R

`{str}` The set of directory paths to add to resources.

:add-source

-S

`{str}` The set of directory paths to add to sources.

:include

-i

`{regex}` The set of regexes that paths must match.

:invert

-v

bool Invert the sense of matching.

:move

-m

`{regex str}` The map of regex to replacement path strings.

:to-asset

-a

`{regex}` The set of regexes of paths to move to assets.

:to-resource

-r

`{regex}` The set of regexes of paths to move to resources.

:to-source

-s

`{regex}` The set of regexes of paths to move to sources.

:with-meta

-w

`{kw}` The set of metadata keys files must have.

:help

bool

Print this help info.

The `--to-asset`, `--to-resource`, and `--to-source` options move matching paths to the corresponding section of the fileset. This can be used to make source files into resource files, for example, etc. If `--invert` is also specified the transformation is done to paths that do ***NOT*** match.

AUTHOR

Written by John Doe.

Related information

- [sift wiki page](#).

target

```
$ boot target --arg ...
```

```
(target :arg ...)
```

DESCRIPTION

Writes output files to the given directory on the filesystem.

where:

:help

bool

Print this help info.

:dir#`{str}`The set of directories to write to (target). Default: `target`**:no-link**

bool

Don't create hard links.

:no-clean

bool

Don't clean target before writing project files.

AUTHOR

Written by John Doe.

Related information

- [target wiki page](#).

uber

```
$ boot uber --arg ...
```

```
(uber :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

• [uber](#) wiki page.

wait

```
$ boot wait --arg ...
```

```
(wait :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

• [show](#) wiki page.

war

```
$ boot war --arg ...
```

```
(war :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [war](#) wiki page.

watch

```
$ boot watch --arg ...
```

```
(watch :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [show](#) wiki page.

web

```
$ boot web --arg ...
```

```
(web :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [web](#) wiki page.

zip

```
$ boot zip --arg ...
```

```
(zip :arg ...)
```

DESCRIPTION

Arguments:

:help

bool

Print this help info.

AUTHOR

Written by John Doe.

Related information

- [zip wiki page](#).

Boot API

The Boot API consists of several namespaces:

boot.core

DESCRIPTION

The boot.core namespace contains functions that ...

AUTHOR

Written by John Doe.

Related information

- [boot.core](#) wiki page.

input-files

DESCRIPTION

AUTHOR

Written by John Doe.

Related information

- [boot.core](#) wiki page.

tmp-dir

DESCRIPTION

AUTHOR

Written by John Doe.

Related information

- [boot.core](#) wiki page.

boot.pod

DESCRIPTION

The boot.pod namespace contains functions that ...

AUTHOR

Written by John Doe.

Related information

- [boot .pod](#) wiki page.

Appendix

This appendix describes things that you rarely need to know.

You can consult this section when you need detailed information about a specific component.

Glossary

AVR

AVR is a kind of microcontroller from Atmel

AVR

Acronym: **AVR**

Related information

- [Atmel AVR 8-bit and 32-bit Microcontrollers](#)

USB flash drive

A small portable drive.

USB flash drive (UFD, or flashie)

Acronym: **UFD**

Explain the acronym on first occurrence.

Arduino

Arduino is ...

Arduino

Related information

- [Arduino](#)

WSN

Wireless Sensor Network. A network of sensor nodes that communicate by radio.

Wireless Sensor Network (WSN)

Acronym: **WSN**

Explain the acronym on first occurrence.

Index

A

add-repo, 7
aot, 7
api
 core, 16
 input-files, 16
 tmp-dir, 16
pod, 16

C

core, 16

F

fileset
 tasks
 add-repo, 7
 aot, 7
 help, 8
 install, 7
 jar, 8
 javac, 9
 pom, 9
 push, 9
 repl, 10
 show, 10
 sift, 11
 uber, 12
 wait, 13
 war, 13
 watch, 14
 web, 14
 zip, 14

H

help, 8

I

input-files, 16
install, 7

J

jar, 8
javac, 9

P

pod, 16
pom, 9

push, 9

R

repl, 10

S

show, 10
sift, 11

T

target, 12
tasks
 built-in
 add-repo, 7
 aot, 7
 help, 8
 install, 7
 jar, 8
 javac, 9
 pom, 9
 push, 9
 repl, 10
 show, 10
 sift, 11
 target, 12
 uber, 12
 wait, 13
 war, 13
 watch, 14
 web, 14
 zip, 14

tmp-dir, 16

U

uber, 12

W

wait, 13
war, 13
watch, 14
web, 14

Z

zip, 14