

# autoproj Cheat Sheet

autoproj >= v1.7.1 / sheet v1.0

## Installing / Config

### Installing autoproj

Installing OS dependencies  
`sudo apt-get install ruby rubygems wget`

Downloading install script  
`wget http://www.rock-robotics.org/autoproj_bootstrap`

Installing autoproj  
`ruby autoproj_bootstrap`

### Bootstrapping projects

Installing OS dependencies  
`sudo apt-get install ruby rubygems wget`

Downloading install script  
`wget http://www.rock-robotics.org/autoproj_bootstrap`

Bootstrapping an existing project  
`ruby autoproj_bootstrap git url branch=all`

### Reconfigure autoproj

re-ask all configuration options  
`autoproj --reconfigure`

switching configurations  
`autoproj switch-config git url branch=all`

Bootstrapping an existing project  
`autoproj bootstrap git url branch=all`

### Monitoring

displaying state of the packages  
`autoproj status`

listing all available packages  
`autoproj list-config`

## Updating / Building

### Updating packages

Sourcing autoproj environment  
`source env.sh`

Updating all packages  
`autoproj update`

Updating configuration only  
`autoproj update-config`

### Building packages

Sourcing autoproj environment  
`source env.sh`

Building all packages + install OS deps  
`autoproj build`

Building all packages without OS deps  
`autoproj fast-build`

Building without stopping on errors  
`autoproj build --keep-going`

Building package xy  
`autoproj build xy`

Installing package related OS deps  
`autoproj osdeps`

Cleaning and building packages  
`autoproj rebuild`

Triggering all build commands  
`autoproj force-build`

Updating and building all packages  
`autoproj full-build`

Generation documentation  
`autoproj doc`

### Overriding config

Overriding source informations  
file: autoproj/overrides.yml  
`overrides:`  
- package\_name:  
url:

## Creating new packages

### Creating new packages

Check list

- create package
- add osdeps to manifest.xml
- add osdeps to local.osdeps or package\_set.osdeps
- add package to autoproj/manifest or package\_set.autobuild and source.yml
- compile new package

`autoproj build`

Creating cmake package  
`rock-create-lib new_package`

Creating new orogen package  
`orogen create package_name`  
(see orogen cheat sheet)

Adding additional package informations  
file: new\_package/manifest.xml

```
<package>
<description brief="">
</description>
<author>Name/Name@xy.org</author>
<license></license>
<url>http://</url>
<logo>http://</logo>
<depend package="" />
<versioncontrol type="git"
url="" />
<export>
<cpp cflags="" lflags="" />
</export>
</package>
```

Overriding build config  
file: autoproj/overrides.rb

## Adding new packages

### Adding packages

Package types

```
cmake_package "name"
ruby_package "name"
orogen_package "name"
autotools_package "name"
import_package "name"
```

Build options

```
not on 'debian' {}
only_on ['ubuntu', '10.04'] {}
cmake_package "name" do |pkg|
  pkg.post_install {}
  pkg.depends_on "pkg_name"
end
```

Adding packages to local config

file: autoproj/local.autoproj  
`cmake_package "name"`  
file: autoproj/manifest  
layout:

- name

Adding packages to a package set  
file: autoproj/remotes/package\_set/  
package\_set.autoproj  
`cmake_package "name"`  
file: autoproj/remotes/package\_set/  
source.yml

version\_control:

- name:
- type: git
- url:
- push\_to:
- branch:

file: autoproj/manifest  
layout:

- name

### Ignoring packages

file: autoproj/manifest

```
exclude_packages:
- package_name
ignore_packages:
- package_name
```

## Defining OS deps

### Adding OS deps

OS deps types

```
name: gem
name:
  debian,ubuntu: [libxy]
name:
  debian:
    squeeze/sid:
      - lib_xy
    gentoo: ignore
```

Adding OS deps to packages

file: package/manifest.xml  
`<depend package = "name"/>`

Defining OS deps for local config

file: autoproj/local.osdeps  
`name: gem`

Defining OS deps for package sets

file: autoproj/remotes/package\_set/  
package\_set.osdeps  
`name: gem`

### Adding package sets

Adding package sets to local config

file: autoproj/manifest  
`package_sets:`  
-type: git  
branch: master  
url:  
push\_to:  
layout:  
- package\_set\_name

Adding package sets to package set

file: autoproj/remotes/package\_set/  
source.yml  
`imports:`  
-type: git  
branch: master  
url:  
push\_to: