Bob BuildTool – Mini Cheat Sheet

Commands

Build

```
bob dev <package query>
bob build <package query>
```

Options

-b	build only mode, no checkout
-c <file>	use additional config file
-DFOO=BAR	define environment variable
-f	force execution
-j<n>	use <n> parallel build slots
-k	keep going
-n	do not build dependencies
-[no-]sandbox	control sandbox
-v[v]	increase verbosity
-resume	resume a interrupted build

List

```
bob ls <package query>
```

Options

-a	show indirect deps too
-c <file>	use additional config file
-d	show packages, not their contents
-D	DEFINES
-o	show origin of indirect packages
-p	print the full prefixed path
-r	recursive

Status

```
bob status [<package query>]
```

Options

-attic	consider attic too
-c <file>	use additional config file
-D	DEFINES
-r	recursive
-v[v]	increase verbosity

Output

A	attic	O	overridden
C	collision	S	switched
E	error	u/U	unpushed
M	modified		

Query Path

```
bob query-path <package query>
```

Options

-c <file>	use additional config file
-D DEFINES	define variable
-f FORMAT	format string
-[no-]sandbox	control sandbox

Format String placeholders

src	checkout directory
build	build directory
dist	package directory

Query Meta / Recipe / SCM

```
bob query-meta <package query>
bob query-recipe <package query>
bob query-scm <package query>
```

Recipe and Class Keywords

checkoutSCM

Type: List of SCMs

```
checkoutSCM:
   scm: [git|url|svn|cvs|import]
   url: <url>
   dir: <local dir in checkout WS>
```

git-scm keywords

branch	Branch to check out
tag	Checkout this tag
commit	SHA1 commit Id to check out
rev	git-rev-parse revision
submodules	control submodules

url-scm keywords

digestSHA1	Expected SHA1
digestSHA256	Expected SHA256
digestSHA512	Expected SHA512
extract	Extract directive
fileName	Local file name

import-scm keywords

prune	Delete dest dir before import

svn-scm keywords

revision	revision number

{checkout,build,package}Script

A String forming the script executed by Bob at the respective Stage.

{checkout,build,package}Tools

List of tools added to the PATH when the script is executed.

{checkout,build,package}Vars

List of environment variables that should be set when the step runs.

Depends

Type :List of Strings or Dependency-Dictionaries

```
depends:
   - name: foo
     use: [tools]
     if: "$(eq,${FOO},42)"
```

Settings

name	name of the dependency
depends	list of dependencies
use	List: [deps, environment, result, tools, sandbox]
checkoutDep	Dep available in checkout step
forward	forward [tools || sandbox || env]
environment	environment for dep
if	use dep if condition is true
tools	remap tools

Environment

Type : Dictionary (String -> String)

```
environment:
   FOO: "BAR"
metaEnvironment:
   LICENSE: "MIT"
privateEnvironment:
   BAR: "BAZ"
```

inherit

Type: List of Strings
Include classes into the current recipe.

multiPackage

Type: Dictionary (String -> Recipe)
Unify multiple recipes into one.

provideDeps

Type: List of patterns
Provided deps are injected into the deplist of the upstream recipe.

provideTools

Type: Dictionary (String -> Path | Tool-Dictionary)
Defines tools that may be used by other steps during the build process.

```
provideTools:
   host-toolchain:
      path: bin
```

Settings

path	relative path
libs	list of library paths added to LD_LIBRARY_PATH
environment	environment defined by the tool

root

Type: Boolean
Defines the package as root package.

User configuration (default.yaml)

environment

Define user environment.

archive

Define bob binary archive (cache).

scmOverrides

Type: List of override specifications.

```
scmOverrides:
   -
      match:
         url: "git@acme.com:/foo/repo.git"
      del: [commit, tag]
      set:
         branch: develop
      replace:
         url:
           pattern: "foo"
            replacement: "bar"
      if: !expr |
         "${BOB_RECIPE_NAME}" == "foo"
```

rootFilter

Filter root-recipes using shell glob patterns.

include

Include additional user config files.