

BitBake cheat sheet

Stable: 19.02.2019 - 12:39 / Revision: 05.02.2019 - 08:56

Contents

1 Command-line options	1
2 .bb file syntax	2
3 Additional BitBake-related command	4
4 Useful links	4

1 Command-line options

Here are a few commonly-used command line options for BitBake.

Option	Command-line options	Meaning
-c <task>		execute <task> for the image or recipe being built. ex: bitbake -c fetch busybox. Some of the possible tasks are: fetch, configure, compile, package, clean
-f		force execution of the operation, even if not required execute tasks from a specific .bb recipe directly. This does not handle any dependencies from other recipes.
-b		<pre>bitbake -b ../meta-st/meta-st-cannes2/recipe</pre>
-v		show verbose output
-DDD		show lots of debug information
-s		show recipe version information output recipe environment variables
-e		<pre>bitbake -e <recipe></pre>
--help		get usage help
-c listtasks <image-or-recipe-name>		show the tasks associated with an image or individual recipe output dependency tree in graphviz format

-g <recipe>

```
bitbake -g <recipe>
dot -v -Tpng -o package-depends.png packag
dot -v -Tpng -o pn-depends.png pn-depends
dot -v -Tsvg -o task-depends.svg task-depe
```

Here are a few examples of command line options for BitBake.

	Main command lines commands	
How to	Command	Comment
How to get the version of packages	bitbake -s	
How to get the Image package list	bitbake <image> -g	Then: cat pn-buildlist
How to clean a module	bitbake <module> -f -c cleanall	
How to clean an image	bitbake <image> -c cleanall	
How to load the packages to a local hard drive	bitbake <image> -f -c fetchall	
How to get the list of tasks	bitbake <image> -f -c listtasks	
How to display all appends	bitbake-layers show-append	
How to display all layers	bitbake <recipe-or-image-name> -g -u depexp	
How to configure the kernel	bitbake <kernel_module> -c menuconfig	kernel module : linux-stm32
How to make an SDK relocatable image	bitbake <image> -c populate_sdk	self-extracting image available in tmp-glibc/deploy/sdk

2 .bb file syntax

This table lists some of the syntaxes found in recipe (.bb) files.

.bb file syntax		
Syntax	Meaning	Notes
VAR = "foo"	simple assignment	
VAR ?= "foo"	assign if no other value is already assigned (default assignment)	
VAR ??=foo	weak default assignment	takes lower precedence than ?=
VAR = "stuff \${OTHER_VAR} more"	variable expansion	OTHER_VAR expanded at time of reference to VAR
		OTHER_VAR expanded when

<code>VAR := "stuff \${OTHER_VAR}"</code>	immediate variable expansion	parsing this line
<code>VAR += "foo"</code>	append with space	
<code>VAR =+ "foo"</code>	prepend with space	
<code>VAR .= "foo"</code>	append without space	
<code>VAR =. "foo"</code>	prepend without space	
<code>VAR_append = "foo"</code>	append without space	
<code>OVERRIDES="string1:string2"</code>	alternate for VAR_value	if string1 is listed in OVERRIDES, use "bar" for value of VAR, otherwise use "foo"
<code>OVERRIDES="string1:string2"</code>	conditional append	if string1 is in OVERRIDES, then append "bar" to the value of VAR
<code>BBVERSIONS="1.0 1.8 string"</code>	range for VAR_override	If the version of the package is in the specified range (1.0-1.8 in this example), then perform an override on the indicated variable
<code>VAR = "foo \${@<line-of-python-code>}"</code>	code expansion	ex: VAR = "the date is: \${@time.strftime('%Y%m%d',time.gmtime())}"
<code>include foo</code>	include file	include file named "foo", search BBPATH
<code>require [<path>]foo</code>	require file	include file named "foo", failing if not found exactly where specified
<code>do_sometask() { <shell code> }</code>	define a task using shell code	

```
python do_sometask {
  <python code>
```

define a task using python code

```
addtask sometask (before|after) @task_task
```

adds a defined task to the list of tasks, with a specific ordering. Zero or more 'before' or 'after' clauses can be used.

a few subsidiary flag value names are well-defined: "dirs", "cleandirs", "noexec", "nostamp", "fakeroot", "umask", "deptask", "rdeptask", "recdeptask", "recrdeptask" %%%

```
VAR [some_flag]="foo"
```

associate a subsidiary flag value to a variable

Flag values appear to be used exclusively with task definitions (i. e. do_sometask)

3 Additional BitBake-related command

Syntax	Meaning	Notes
bitbake-layers	Shows information about layers and recipes	Included in the bitbake/bin directory in openembedded source tree.
bitbake-whatchanged <image>	Shows all changes made since the last generation of image	<pre>bitbake st-image-weston # Edit the recipes bitbake-whatchanged st-image</pre>
bitbake -g -u depexp <recipe-or-image-name>	Shows dependency information in a graphical interface	This is a simple graphical wrapper over the streaming text output of BitBake. It shows collapsible trees for the logs for sub-tasks for each recipe.
bitbake -u goggle st-image-weston	simple graphical interface	

4 Useful links

- <http://docs.openembedded.org/bitbake/html/>
- http://www.openembedded.org/wiki/Main_Page
- <http://www.yoctoproject.org/docs/1.6/dev-manual/dev-manual.html>
- <http://www.yoctoproject.org/docs/1.6/bitbake-user-manual/bitbake-user-manual.html>

Software development kit

