

## When trying to build RDK code,

I'm getting this error:

ERROR: Error executing a python function in /home/rdkm-17.2/meta-rdk/recipes-core/images/[rdk-generic-broadband-image.bb](#):

The stack trace of python calls that resulted in this exception/failure was:

```
File: 'do_rootfs', lineno: 17, function: <module>
0013: # generate final images
0014: create_image(d)
0015:
0016:
*** 0017:do_rootfs(d)
0018:
File: 'do_rootfs', lineno: 11, function: do_rootfs
0007: # generate the initial manifest
0008: create_manifest(d)
0009:
0010: # generate rootfs
*** 0011: create_rootfs(d)
0012:
0013: # generate final images
0014: create_image(d)
0015:
File: '/home/rdkm-17.2/openembedded-core/meta/lib/oe/rootfs.py', lineno: 790, function: create_rootfs
0786: img_type = d.getVar('IMAGE_PKGTYPE', True)
0787: if img_type == "rpm":
0788: RpmRootfs(d, manifest_dir).create()
0789: elif img_type == "ipk":
*** 0790: OpkgRootfs(d, manifest_dir).create()
0791: elif img_type == "deb":
0792: DpkgRootfs(d, manifest_dir).create()
0793:
0794: os.environ.clear()
File: '/home/rdkm-17.2/openembedded-core/meta/lib/oe/rootfs.py', lineno: 158, function: create
0154: bb.utils.mkdrihier(sysconfdir)
0155: with open(sysconfdir + "/version", "w+") as ver:
0156: ver.write(self.d.getVar('BUILDNAME', True) + "\n")
0157:
*** 0158: self._run_intercepts()
0159:
0160: execute_pre_post_process(self.d, post_process_cmds)
0161:
0162: if base_contains("IMAGE_FEATURES", "read-only-rootfs",
File: '/home/rdkm-17.2/openembedded-core/meta/lib/oe/rootfs.py', lineno: 236, function: _run_intercepts
0232:
0233: bb.note("> Executing %s intercept ..." % script)
0234:
0235: try:
*** 0236: subprocess.check_output(script_full)
0237: except subprocess.CalledProcessError as e:
0238: bb.warn("The postinstall intercept hook '%s' failed (exit code: %d)! See log for details!" %
0239: (script, e.returncode))
0240:
File: '/usr/lib/python2.7/subprocess.py', lineno: 537, function: check_output
0533: 'ls: non_existent_file: No such file or directory\n'
0534: ""
0535: if 'stdout' in kwargs:
0536: raise ValueError('stdout argument not allowed, it will be overridden.')
```

```
*** 0537: process = Popen(stdout=PIPE, *popenargs, **kwargs)
0538: output, unused_err = process.communicate()
0539: retcode = process.poll()
0540: if retcode:
0541: cmd = kwargs.get("args")
File: '/usr/lib/python2.7/subprocess.py', lineno: 679, function: __init__
0675: cwd, env, universal_newlines,
0676: startupinfo, creationflags, shell,
0677: p2cread, p2cwrite,
0678: c2pread, c2pwrite,
*** 0679: errread, errwrite)
0680:
0681: if mswindows:
0682: if p2cwrite is not None:
0683: p2cwrite = msvcrt.open_osfhandle(p2cwrite.Detach(), 0)
File: '/usr/lib/python2.7/subprocess.py', lineno: 1249, function: _execute_child
1245: child_exception = pickle.loads(data)
1246: for fd in (p2cwrite, c2pread, errread):
1247: if fd is not None:
1248: os.close(fd)
*** 1249: raise child_exception
1250:
1251:
1252: def _handle_exitstatus(self, sts, _WIFSIGNALED=os.WIFSIGNALED,
1253: _WTERMSIG=os.WTERMSIG, _WIFEXITED=os.WIFEXITED,
Exception: OSError: [Errno 13] Permission denied
```

DEBUG: Python function do\_rootfs finished  
ERROR: Function failed: do\_rootfs

Sadly, I don't see enough info in this error trace to really tell me  
what file/directory is permission denied (i.e. what the real problem is)

Note: This error came out if I import the code to the SVN server then check it out use another build code host, but if I use the build code host that import to  
checkout the code, it can build the code successfully.