[yocto] The simplest possible recipe

Paul Eggleton paul.eggleton at linux.intel.com
Tue Jun 25 03:17:47 PDT 2013

- Previous message: [yocto] The simplest possible recipe
- Next message: [yocto] The simplest possible recipe
- **Messages sorted by:** [date] [thread] [subject] [author]

Hi Paul,

On Monday 24 June 2013 17:18:05 Paul D. DeRocco wrote:
> *That would be to copy a single file, provided in the files subdirectory of*
> *the recipe, into a particular place in the target tree. Is there any bbclass*
> *that automates this? Or do I just write a recipe with a do_install function*
> *that executes the "install" command?*

There's nothing that automates this, no. On the face of it it seems straightforward, but when installing you need to know what the permissions should be, you may wish to name the file differently than it is named in SRC_URI, etc. For the few cases were a recipe does need to do this it's easy enough to just get the recipe to do everything (which is not a lot).

> *My only guide is 5.3.1 in the Development Manual, which performs a simple*
> *compilation, but I'm very hazy about how recipes are interpreted, so I'd*
> *like it if someone can tell me if I've gotten the following stuff right or*
> *not.*
>
> *SRC_URI tells bitbake what files must be gotten from somewhere and copied*
> *somewhere else in order to carry out the build process.*

To the work directory (WORKDIR) yes.

> *And according to the Ref Manual, the "file://" prefix tells it to fetch a*
> *local file by searching some directories including the "files" subdirectory*
> *next to the .bb file.*

It searches FILESPATH for local files, technically, which by default includes subdirectories named "files" and the bare name of the recipe (${BPN}) as well as the bare name with the version separated by a dash (${BP}). So for a recipe called example-software_2.5.bb it would search "files", "example-software", and "example-software-2.5".

> *And apparently, there is a "subdir" option (whose syntax is unexplained)*
> *which may be used to tell bitbake to put it somewhere specific*
> *relative to ${WORKDIR}.*
>
> *Is the default value of the "subdir" option the S variable?*

No. By default subdir is empty - since the convention for tarballs in free/open source software is to have a subdirectory inside the tarball already, we don't need to extract the contents into one explicitly, we just extract it under the WORKDIR and the contents will already be under a subdirectory. The subdir parameter exists for those upstream tarballs where for whatever reason the creator of the tarball has not included a subdirectory, or has used a subdirectory which clashes with another subdirectory name within the workdir that we already use ("image", "packages", etc.) If you do set subdir you'll also need to set S to match it.

> *Is that the purpose of S, to tell bitbake where to put things that it*

> *fetches?*

Rather, it tells the build system where to find the sources after they have
been unpacked.

> *The Ref Manual says that S defaults to ${WORKDIR}/${PN}/${PV}*

${WORKDIR}/${BP} is the default (equivalent to ${WORKDIR}/${BPN}-${PV}). This
is a common convention based on the naming of upstream tarballs, but is by no
means always the right value, hence you'll see a lot of recipes set S to the
correct value to point to whatever will be extracted from the upstream tarball
fetched by that recipe.

> *then the sample compile recipe sets S to ${WORKDIR}. Is that what one does*
> *when one doesn't need to have a bunch of versioned subdirectories under*
> *${WORKDIR}? (I'm not sure why one would ever want that, or why that would be*
> *the default.)*

If you know you're not going to unpack an archive into a subdirectory, and for
the single C file you aren't, then S can be set to ${WORKDIR} because that's
where the C file will be copied to.

> *So if I want to install a file somewhere, do I even need a do_install task,*
> *or can I just set S equal to the desired target location, like*
> *"${etcdir}/foo" and be done with it? Or is that a no-no, and should I always*
> *use do_install?*

You should always use do_install. You'll lose a bunch of flexibility trying to
do it the other way and you may break some assumptions the system makes about
what it can do with S, not to mention that the sstate code might break.

HTH.

Cheers,
Paul

--

Paul Eggleton
Intel Open Source Technology Centre

- Previous message: [yocto] The simplest possible recipe
- Next message: [yocto] The simplest possible recipe
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the yocto mailing list