# Methods: Inbuilt methods to make your life easier

There are several helper functions defined by the base class, which is included by default for all recipes. Many of these are used a lot in both recipes and other classes.

The most commonly seen, and most useful functions, include:

oe_runmake

This function is used to run make. However unlike calling make yourself this will pass the EXTRA_OEMAKE settings to make, will display a note about the make command and will check for any errors generated via the call to make.

You should never have any reason to call make directly and should also use oe_runmake when you need to run make.

oe_runconf (autotools only)

This function is used to run the configure script of a package that is using the autotools class. This takes care of passing all of the correct parameters for cross-compiling and for installing into the appropriate target directory.

It also passes the value of the **EXTRA_OECONF** variable to the configure script. For many situations setting **EXTRA_OECONF** is sufficient and you'll have no need to define your own configure task in which you call oe_runconf manually.

If you need to write your own *configure* task for an autotools package you can use oe_runconf to manually call the configure process when it is required. The following example from net-snmp shows oe_runconf being called manually so that the parameter for specifying the endianess can be computed and passed in to the configure script:

```
do_configure() {
    # Additional flag based on target endiness (see siteinfo.bbclass)
    ENDIANESS="${@base_conditional('SITEINFO_ENDIANESS', 'le', '--with-endianness=little', '--with-endianness=big', d)}"
    oenote Determined endianess as: $ENDIANESS
    oe_runconf $ENDIANESS
}
```

oe_libinstall

This function is used to install **.so**, **.a** and associated libtool **.la** libraries. It will determine the appropriate libraries to install and take care of any modifications that may be require for **.la** files.

This function supports the following options:

-C

Change into the specified directory before attempting to install a library. Used when the libraries are in subdirectories of the main package.

-s

Require the presence of a **.so** library as one of the libraries that is installed.

-a

Require the presence of a **.a** library as one of the libraries that is installed.

The following example from gdbm shows the installation of **.so**, **.a** (and associated **.la**) libraries into the staging library area:

```
do_stage () {
    oe_libinstall -so -a libgdbm ${STAGING_LIBDIR}
    install -m 0644 ${S}/gdbm.h ${STAGING_INCDIR}/
}
```

oenote

Used to display an informational messages to the user.

The following example from net-snmp uses oenote to tell the user which endianess it determined was appropriate for the target device:

```
do_configure() {
    # Additional flag based on target endiness (see siteinfo.bbclass)
    ENDIANESS="${@base_conditional('SITEINFO_ENDIANESS', 'le', '--with-endianness=little', '--with-endianness=big', d)}"
    oenote Determined endianess as: $ENDIANESS
    oe_runconf $ENDIANESS
}
```

oewarn

Used to display a warning message to the user, warning of something that may be problematic or unexpected.

oedebug

Used to display debugging related information. These messages will only be visible when bitbake is run with the **-D** flag to enable debug output.

oefatal

Used to display a fatal error message to the user, and then abort the bitbake run.

The following example from linux-libc-headers shows the use of oefatal to tell the user when it cannot find the kernel source code for the specified target architecture:

```
do_configure () {
    case ${TARGET_ARCH} in
        alpha*)   ARCH=alpha ;;
        arm*)     ARCH=arm ;;
        cris*)    ARCH=cris ;;
        hppa*)    ARCH=parisc ;;
        i*86*)    ARCH=i386 ;;
        ia64*)    ARCH=ia64 ;;
        mips*)    ARCH=mips ;;
        m68k*)    ARCH=m68k ;;
        powerpc*) ARCH=ppc ;;
        s390*)    ARCH=s390 ;;
        sh*)      ARCH=sh ;;
        sparc64*) ARCH=sparc64 ;;
        sparc*)   ARCH=sparc ;;
        x86_64*)  ARCH=x86_64 ;;
    esac
    if test !  -e include/asm-$ARCH; then
        oefatal unable to create asm symlink in kernel headers
    fi
...
```

base_conditional (python)

The base conditional python function is used to set a variable to one of two values based on the definition of a third variable. The general usage is:

```
${@base_conditional('', '', '', ', d)}"
```

where:

variable-name

This is the name of a variable to check.

value

This is the value to compare the variable against.

true-result

If the variable equals the value then this is what is returned by the function.

false-result

If the variable does not equal the value then this is what is returned by the function.

### Note

The ${@...} syntax is used to call python functions from within a recipe or class. This is described in more detail in the advanced python section.

The following example from the openssl recipe shows the addition of either **-DL_ENDING** or **-DB_ENDIAN** depending on the value of **SITEINFO_ENDIANESS** which is set to le for little endian targets and to be for big endian targets:

```
do_compile () {
    ...
    # Additional flag based on target endiness (see siteinfo.bbclass)
    CFLAG="${CFLAG} ${@base_conditional('SITEINFO_ENDIANESS', 'le', '-DL_ENDIAN', '-DB_ENDIAN', d)}"
    ...
```

Prev Up Next
Dependencies: What's needed to build and/or run the package? Home Packaging: Defining packages and their contents