**2.4. Working With Your Own Sources[¶](http://www.yoctoproject.org/docs/2.3.1/kernel-dev/kernel-dev.html" \l "working-with-your-own-sources" \o "Permalink)**

If you cannot work with one of the Linux kernel versions supported by existing linux-yocto recipes, you can still make use of the Yocto Project Linux kernel tooling by working with your own sources. When you use your own sources, you will not be able to leverage the existing kernel [Metadata](http://www.yoctoproject.org/docs/2.3.1/dev-manual/dev-manual.html#metadata) and stabilization work of the linux-yocto sources. However, you will be able to manage your own Metadata in the same format as the linux-yocto sources. Maintaining format compatibility facilitates converging with linux-yocto on a future, mutually-supported kernel version.

To help you use your own sources, the Yocto Project provides a linux-yocto custom recipe (linux-yocto-custom.bb) that useskernel.org sources and the Yocto Project Linux kernel tools for managing kernel Metadata. You can find this recipe in the poky Git repository of the Yocto Project [Source Repository](http://git.yoctoproject.org/) at:

poky/meta-skeleton/recipes-kernel/linux/linux-yocto-custom.bb

Here are some basic steps you can use to work with your own sources:

1. Copy the linux-yocto-custom.bb recipe to your layer and give it a meaningful name. The name should include the version of the Linux kernel you are using (e.g. linux-yocto-myproject\_3.19.bb, where "3.19" is the base version of the Linux kernel with which you would be working).
2. In the same directory inside your layer, create a matching directory to store your patches and configuration files (e.g. linux-yocto-myproject).
3. Make sure you have either a defconfig file or configuration fragment files. When you use the linux-yocto-custom.bbrecipe, you must specify a configuration. If you do not have a defconfig file, you can run the following:
4. $ make defconfig

After running the command, copy the resulting .config to the files directory as "defconfig" and then add it to the [SRC\_URI](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-SRC_URI)variable in the recipe.

Running the make defconfig command results in the default configuration for your architecture as defined by your kernel. However, no guarantee exists that this configuration is valid for your use case, or that your board will even boot. This is particularly true for non-x86 architectures. To use non-x86 defconfig files, you need to be more specific and find one that matches your board (i.e. for arm, you look in arch/arm/configs and use the one that is the best starting point for your board).

1. Edit the following variables in your recipe as appropriate for your project:
   * [SRC\_URI](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-SRC_URI): The SRC\_URI should specify a Git repository that uses one of the supported Git fetcher protocols (i.e. file, git,http, and so forth). The SRC\_URI variable should also specify either a defconfig file or some configuration fragment files. The skeleton recipe provides an example SRC\_URI as a syntax reference.
   * [LINUX\_VERSION](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-LINUX_VERSION): The Linux kernel version you are using (e.g. "3.19").
   * [LINUX\_VERSION\_EXTENSION](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-LINUX_VERSION_EXTENSION): The Linux kernel CONFIG\_LOCALVERSION that is compiled into the resulting kernel and visible through the uname command.
   * [SRCREV](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-SRCREV): The commit ID from which you want to build.
   * [PR](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-PR): Treat this variable the same as you would in any other recipe. Increment the variable to indicate to the OpenEmbedded build system that the recipe has changed.
   * [PV](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-PV): The default PV assignment is typically adequate. It combines the LINUX\_VERSION with the Source Control Manager (SCM) revision as derived from the [SRCPV](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-SRCPV) variable. The combined results are a string with the following form:
   * 3.19.11+git1+68a635bf8dfb64b02263c1ac80c948647cc76d5f\_1+218bd8d2022b9852c60d32f0d770931e3cf343e2

While lengthy, the extra verbosity in PV helps ensure you are using the exact sources from which you intend to build.

* + [COMPATIBLE\_MACHINE](http://www.yoctoproject.org/docs/2.3.1/ref-manual/ref-manual.html#var-COMPATIBLE_MACHINE): A list of the machines supported by your new recipe. This variable in the example recipe is set by default to a regular expression that matches only the empty string, "(^$)". This default setting triggers an explicit build failure. You must change it to match a list of the machines that your new recipe supports. For example, to support theqemux86 and qemux86-64 machines, use the following form:
  + COMPATIBLE\_MACHINE = "qemux86|qemux86-64"

1. Provide further customizations to your recipe as needed just as you would customize an existing linux-yocto recipe. See the "[Modifying an Existing Recipe](http://www.yoctoproject.org/docs/2.3.1/kernel-dev/kernel-dev.html#modifying-an-existing-recipe)" section for information.