

Proposal for  
`--easyblock=CLASS --help`

# Mock-up

- `eb --easyblock=EB_FSL --help`
- `eb --easyblock-help=EB_FSL <=` easier to implement (check `tools_options.py`)

FSL:

Support for building and installing FSL. Derives from EasyBlock.

`configure_step`:

Sets the `FSL_DIR` environment variable to the starting dir.

FSL `<= 6.0.1`: FSL ships with hardcoded template makefiles for various architectures. The `configure_step` checks that at least a single patch file in the `easyconfig` edits a template makefile. If no patched config is found, a best guess is made to select one of the template makfiles.

FSL `>= 6.0.2`: FSL only ships a generic template makefile. A patch needs to be used to edit this makefile. The `configure_step` does not verify if such a patch was supplied.

`Build_step`:

Sources the `etc/fslconf/fsl.sh` script and runs FSL's *build* script.

`Sanity_check_step`:

Checks presense of various folders in the FSL installation directory. Then calls the sanity check of the parent class.

# Mock-up

- `eb --easyblock=EB_FSL -r --help`

FSL:

Support for building and installing FSL. Derives from EasyBlock.

`configure_step`:

Sets the `FSL_DIR` environment variable to the starting dir.

FSL <= 6.0.1: FSL ships with hardcoded template makefiles for various architectures. The `configure_step` checks that at least a single patch file in the `easyconfig` edits a template makefile. If no patched config is found, a best guess is made to select one of the template makfiles.

FSL >= 6.0.2: FSL only ships a generic template makefile. A patch needs to be used to edit this makefile. The `configure_step` does not verify if such a patch was supplied.

`Build_step`:

Sources the `etc/fslconf/fsl.sh` script and runs FSL's *build* script.

`Sanity_check_step`:

Checks presense of various folders in the FSL installation directory. Then calls the sanity check of the parent class.

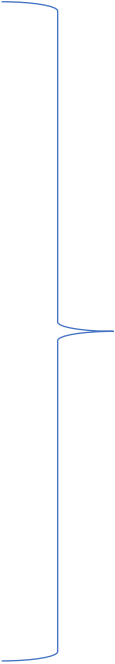
`Permissions_step`:

<prints `easyblock.permissions_step.__doc__`>

# Points of interest

- The 2nd is easier to create: Python introspection does it by default

```
class EasyBlock:
    """This is the EasyBlock class description"""
    def configure(self):
        """This is the configure step of EasyBlock"""
    def install(self):
        """This is the install step of EasyBlock"""
class SpecificEasyBlock(EasyBlock):
    """This is the SpecificEasyBlock class description"""
    def configure(self):
        """This is the configure step of SpecificEasyBlock"""
        super(SpecificEasyBlock, self).configure()
```



```
> SpecificEasyBlock.configure.__doc__
> 'This is the configure step of SpecificEasyBlock'

> SpecificEasyBlock.install.__doc__
> 'This is the install step of EasyBlock'
```

# Points of interest

- The 2nd may produce too much info to be useful: all the docstrings from EasyBlock!
- Is there a way to only print docstrings of the current child-class? Would that be (more) useful?

# Technical implementation

```
class EasyBlock:
    """This is the EasyBlock class description"""
    def configure(self):
        """This is the configure step of EasyBlock"""
    def build(self):
        """This is the build step of EasyBlock"""
    def install(self):
        """This is the install step of EasyBlock"""
class SpecificEasyBlock(EasyBlock):
    """This is the SpecificEasyBlock class description"""
    def configure(self):
        """This is the configure step of SpecificEasyBlock"""
        super(SpecificEasyBlock, self).configure()
    def build(self):
        """This is the build step of SpecificEasyBlock"""
import pydoc
print(pydoc.render_doc(SpecificEasyBlock, renderer=pydoc.plaintext))
```

# Would print...

```
[casparl@int2 TMP]$ python test.py
Python Library Documentation: class SpecificEasyBlock in module __main__

class SpecificEasyBlock(EasyBlock)
| This is the SpecificEasyBlock class description
...
| Methods defined here:
| build(self)
|     This is the build step of SpecificEasyBlock
| configure(self)
|     This is the configure step of SpecificEasyBlock
| -----
| Methods inherited from EasyBlock:
|
| install(self)
|     This is the install step of EasyBlock
...
```

# Different format from eb --help...

- Quite ugly, but it might require a lot of effort to put it in a different format (?)
- Maybe easier for someone already familiar with the help parsers of current EasyBuild...