

Proposal for `--tweak-deps`

We have freetype-2.9.1-GCCcore-7.3.0.eb

```
name = 'freetype'  
version = '2.9.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '7.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

# We want freetype-2.10.1-GCCcore-8.3.0.eb

```
eb -D freetype-2.9.1-GCCcore-7.3.0.eb --try-software-version=2.10.1 --  
disable-cleanup-tmpdir
```

```
name = 'freetype'  
version = '2.9.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '7.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

```
name = 'freetype'  
version = '2.10.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '7.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

# We want freetype-2.10.1-GCCcore-8.3.0.eb

```
eb -D freetype-2.10.1-GCCcore-7.3.0.eb --try-toolchain=version=8.3.0 -  
-disable-cleanup-tmpdir
```

```
name = 'freetype'  
version = '2.10.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '7.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

```
name = 'freetype'  
version = '2.10.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '8.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

# We want freetype-2.10.1-GCCcore-8.3.0.eb

```
eb -D freetype-2.10.1-GCCcore-8.3.0.eb --tweak-deps=deplist.txt --  
disable-cleanup-tmpdir
```

```
name = 'freetype'  
version = '2.10.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '8.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.30'),  
]  
dependencies = [  
    ('bzip2', '1.0.6'),  
    ('libpng', '1.6.34'),  
    ('zlib', '1.2.11'),  
]  
...
```

```
name = 'freetype'  
version = '2.10.1'  
...  
toolchain = {'name': 'GCCcore', 'version': '8.3.0'}  
...  
builddependencies = [  
    ('binutils', '2.32'),  
]  
dependencies = [  
    ('bzip2', '1.0.8'),  
    ('libpng', '1.6.37'),  
    ('zlib', '1.2.11'),  
]  
...
```

# What should deplist.txt look like?

Option 1: replace any occurrence of ('binutils', \*) with ('binutils', '2.32')

```
# deplist.txt  
[  
  'binutils', '2.32'  
  'bzip2': '1.0.8'  
  'libpng': '1.6.37'  
]
```

- Question: how should this behave for suffixes?

('PyCairo', '1.18.0', 'Python-2.7.15'), => ('PyCairo', '1.20.0', 'Python-2.7.15') => BAD!

('PyCairo', '1.18.0', versionsuffix), => ('PyCairo', '1.20.0', versionsuffix) => OK! (probably)

- Simply issue a warning/error if versionsuffix is a string? (--try-\* is also not fool-proof)

# What should deplist.txt look like?

## Option 2: define replacements explicitly

```
# deplist.txt
[
  (('binutils', '2.30'), ('binutils', '2.32')),
  ...
  (('PyCairo', '1.18.0', 'Python-2.7.15'), ('PyCairo', '1.20.0', 'Python-2.7.16'))
]
```

- Question: how should this handle templates?

('PyCairo', '1.18.0', 'Python-2.7.15'), => ('PyCairo', '1.20.0', 'Python-2.7.16') => OK!

('PyCairo', '1.18.0', versionsuffix), => ('PyCairo', '1.20.0', ???) => OK! (probably)

- Requires many more definitions... (this example: also for Python-3.\*)

# What should deplist.txt look like?

- Option 3: combination of (1) and (2)

```
# deplist.txt  
[  
  ('binutils', '2.32'),           Replace any binutils by version 2.32  
  ...  
  (('PyCairo', '1.18.0', 'Python-2.7.15'), ('PyCairo', '1.20.0', 'Python-2.7.16'))  
]
```

# Proposal: minimal effort, maximum gain

- Go for Option 1
- Replaces any version, i.e.

('binutils', '2.30'),	=> ('binutils', '2.32')	=> OK!
('PyCairo', '1.18.0', 'Python-2.7.15'),	=> ('PyCairo', '1.20.0', 'Python-2.7.15')	=> BAD!
('PyCairo', '1.18.0', versionsuffix),	=> ('PyCairo', '1.20.0', versionsuffix)	=> OK! (probably)
('PyCairo', '1.18.0', "", ('GCCcore', '7.3.0')),	=> ('PyCairo', '1.20.0', "", ('GCCcore', '7.3.0'))	=> BAD!

- Useful for 95% of EasyConfigs
- Prints clear warning (Error?) when suffix / toolchain is specified for dependency
- Still allows potential to be extended to Option 3 later on

# Remaining questions

- Should it **ALSO** work on the version? (i.e. replace `--try-software-version`)
  - That would make sense: only having the `deplist.txt` would then allow bumping *all* versions (software *and* deps) when moving to next toolchain.
  - `--tweak-deps` then not the best name?
- Suggestion for `deplist.txt` format?
- Who can help me find my way in `easybuild-framework`?
  - Point out probable locations for changes
  - Point out existing tools that may help