Building and packaging FAQ

Deprecation warning

The page has been migrated to https://github.com/OpenIndiana/oi-userland/blob/oi/hipster/doc/builidng-and-packaging-faq.md . Avoid touching this wiki page.

This page contains answers to the most common questions regarding building and packaging software on OpenIndiana.

How to bypass checks for .pyc files in IPS manifests?

Generally there might be cases when we don't want to deliver .pyc in manifest. In that case you should append **pkg.tmp.autopyc=false**, e.g:

```
file path=usr/lib/Modules/$(COMPONENT_VERSION)/init/python.py pkg.tmp.autopyc=false
```

If this needs to be achieved for the whole IPS manifest, it's better to use a **transform**, e.g

```
<transform file path=.*\.py$ -> default pkg.tmp.autopyc false>
```

Why compiled python files are no longer automatically added to manifests?

After introduction of Python 3.4 autopyc transform has to distinguish compiled files for python 2.x (*.pyc) and for python 3.x (__pycache__/*.cpython-3x.pyc). This is done on the base of file path.
Files in /usr/lib/python2.x/*.py are processed as earlier (following Python 2.x convention). Files in /usr/lib/python3.x are processed according to Python 3.x convention. All other *.py files are not
processed automatically.

Why do I receive "mv: cannot access .deps/libsomething.Tpo" error during package build?

The most frequent reason for this error is a regeneration of Makefile.in files in the wrong directory (when you used $(CLONEY) to populate build directory with sources and then run automake). Generally, libtoolize, aclocal, automake and friends should be run as COMPONENT_PREP_ACTION when $(@D) is $(SOURCE_DIR), before changing $(@D) to $(BUILD_DIR_$(BITS)). $(CLONEY) should be run in COMPONENT_PRE_CONFIGURE_ACTION. Another possible reason is missing some step in configuration - e.g. libtoolize.

Why COMPONENT_AUTOGEN_MANIFEST is dangerous?

Setting COMPONENT_AUTOGEN_MANIFEST=yes in your component's Makefile and adding empty *.p5m manifest you can make oi-userland build system to generate *.p5m manifest automatically. But it's not a very good idea: it hides manifest from you and so manifest can contain garbage. Even more, if you change component or its dependencies, the files can be missing from the package without any warning. The better alternative is to use "gmake sample-manifest" and modify generated manifest so that it satisfies your needs.