

MaNifestO — the oinm manifesto

Mark Weyer

May 30, 2012

1 Oinm is not make. Why not?

Oinm is a build tool. Make is a well-established build tool. So why would one want to use oinm? When using make, the author was unsatisfied in a number of ways:

1. Changing a rule in a **Makefile** does not cause its target to be rebuilt. Yet such a change affects the target in much the same way as changing one of its sources.

With oinm, each rule has an implicit dependency on itself. That is, if the rule changes, its targets will be rebuilt.

2. In a chain of rules, all rules are executed when a source was changed (or even only **touched**), even if the intermediate files are the same after their rebuild as before, for example because the only change in the source was an additional comment. Such superfluous execution of rules can become very painful when the later rules in the chain are expensive.

With oinm, whether a target is up to date is not determined by its timestamp but instead by its content. More precisely a hash of its content, but if you are paranoid you can choose **cat** as the hash command (in future versions of oinm, that is).

3. There is no easy way to autogenerate rules.

There is a wildcard mechanism, but it does not work well when different rules for the same type of target require different command line options.

There is the possibility to include sub-makefiles, but it pollutes the working directory to do so extensively.

Oinm has a Turing-complete mechanism for generating rules.

4. When working on the same sources from different machines, clock skews regularly cause make to do the wrong thing. Also, clock granularity is an issue when a rule needs less than a second to complete.

As noted above, oinm does not use timestamps.

5. Some rules produce more than one target at once. An example is latex, which turns a `.tex` into a `.dvi`, a `.aux`, a `.log`, and possibly others depending on the used packages. Make does not seem¹ to support specifying such rules.

Oinm rules can have multiple targets.

6. There is no free documentation for make.

This MaNifestO is under the same license as oinm is.

The remainder of this document is less of a manifesto and more of a manual.

2 How does it work?

Oinm is controled by an `oinmfile`. The `oinmfile` contains rules for generating targets from sources and rules for generating parts of itself from sources. Recursively executing the latter type of rules results in an expanded `oinmfile` which contains all rules of the former type.

3 Invocation of oinm

4 Syntax of oinmfile

5 Oinm tools

5.1 oinm4ocaml

¹See the next point