## NAME

autopod − generating documentation from POD formatted input with the MST autotool setup

## DESCRIPTION

POD formatted text is used in a number of places. There are a number of common usage scenarios which have been codified into easy to use snippets. More complex usage requires a more in-depth understanding of how the POD snippets tie together.

In order to ensure that the exported distribution does not depend upon the tools required to generate the documentation, the formatted output must be distributed along with any intermediate dependencies. The documentation may have values interpolated into during the maintainer's build process, but those values will be frozen and shipped. To prevent confusion, make sure that those values are fairly benign (like the package version), and try to avoid putting absolute paths into the documentation.

Future updates to the snippets may check for the presence of the documentation tools and allow generation of output upon their presence.

The standard snippets install the documentation in the following places:

```
$(prefix)/share/<prog>/html/<prog>.html
$(prefix)/share/<prog>/<prog>.ps
$(prefix)/man/manl/<prog>.l
```

## USAGE

*configure.ac*

Add

```
m4_include(snippets/pod_docs.ac)
```

*Makefile.am*

If the documentation is in the *docs* subdirectory and is interpolated (via *configure.ac* or *docs/Makefile.am*) ensure that the *docs* directory *preceeds* the source code directory in the `SUBDIRS` list so that the interpolation is done before the source build process requires it.

### Interpolating into POD source

If the POD source is interpolated into (either by *configure* or by a rule in the *Makefile*) substitute

```
include $(top_srcdir)/snippets/prog_pod_subst.mk
```

for

```
include $(top_srcdir)/snippets/prog_pod.mk
```

and

```
include $(top_srcdir)/snippets/lib_pod_subst.mk
```

for

```
include $(top_srcdir)/snippets/lib_pod.mk
```

in the directions below. The input POD files must have an additional *.in* suffix:

```
foo.pod.in
```

### Common Scenarios

*Perl Executable Source With Embedded Documentation*

If the POD is embedded into a single Perl source file (as is the case with most Perl applications), say *<prog>.pl.in*, then the following magic may be invoked in *Makefile.am*

```
include $(top_srcdir)/snippets/create_am_macros.mk
include $(top_srcdir)/snippets/prog_perl.mk

dist_manl_MANS = <prog>.l
POD_SFX = .pl.in
PODS = <prog>
```

```
include $(top_srcdir)/snippets/prog_pod.mk
```

If multiple source files have documentation, simply append them to the `dist_manl_MANS` and `PODS` variables. Note that

- The files must all have the same suffix (POD_SFX)
- `dist_manl_MANS` must be an explicit list; it cannot use any **make** variable substitutions.

*A Program with Standalone Documentation*

Here the POD source is in a separate POD file. Insert the following in *Makefile.am*

```
include $(top_srcdir)/snippets/create_am_macros.mk

dist_manl_MANS = <prog>.l
POD_SFX = .pod
PODS = <prog>

include $(top_srcdir)/snippets/prog_pod.mk
```

If the suffix is other than `.pod`, amend the above to reflect that.

*A Library with Standalone Documentation*

Here the POD source is in a separate POD file. Insert the following in *Makefile.am*

```
include $(top_srcdir)/snippets/create_am_macros.mk

dist_man3_MANS = <lib>.3
POD_SFX = .pod
PODS = <lib>

include $(top_srcdir)/snippets/lib_pod.mk
```

If the suffix is other than `.pod`, amend the above to reflect that.

*Multiple Manual Pages Intended For Different Sections*

If there are multiple manual pages destined for different sections of the UNIX manual (e.g, `lib.3` and `lib.5`), things are a little different. *All* POD source files must have names which include the section identifier:

```
      <prog>.l.pod
      <lib>.3.pod
      <file>.5.pod
```

Only sections `l`, `3`, and `5` are supported.

Insert the following into *Makefile.am*

```
include include $(top_srcdir)/snippets/create_am_macros.mk

dist_manl_MANS = <prog>.l
dist_man3_MANS = <lib>.3
dist_man5_MANS = <file>.5
POD_SFX = .pod
PODS = <prog>.l <lib>.3 <lib>.5

include include $(top_srcdir)/snippets/pod_all.mk
```

Obviously, only include the `dist_manX_MANS` variables which are required.

## More Complex Scenarios

*Generating only man pages*

If only a man page is desired (for instance if the main documentation is in texinfo, which does not produce a man page), it is necessary to circumvent the construction of HTML and PostScript. This requires including only those parts of the *pod_man* structure that are required. Here's the magic for *Makefile.am* for a section `l` manual page derived from a file called <*prog*>*.pod*:

```
include include $(top_srcdir)/snippets/create_am_macros.mk

# this part is standard
dist_manl_MANS = <prog>.l
POD_SFX = .pod
PODS    = <prog>

# need these because we call pod_man directly
POD_DIR = $(srcdir)/
EXTRA_DIST += $(PACKAGE_NAME)$(POD_SFX)

include include $(top_srcdir)/snippets/pod_man_init.mk
include include $(top_srcdir)/snippets/pod_manl.mk
```