Using XMOS Makefiles to create binary libraries

IN THIS DOCUMENT

- The module_build_info file
- The module Makefile
- Using the module

The default module system used by XMOS application makefiles includes common modules at the source code level. However, it is possible to build a module into a binary library for distribution without the source.

A module that is to be built into a library needs to be split into source that is used to build the library and source/includes that are to be distributed with the library. For example, you could specify the following structure.

```
module_my_library/
       Makefile
       module_build_info
       libsrc/
          my_library.xc
       src/
          support_fns.xc
       include/
          my_library.h
```

The intention with this structure is that the source file `my_library.xc` is compiled into a library and that library will be distributed along with the `src` and `include` directories (but not the `libsrc` directory).

1 The module_build_info file

To build a binary library some extra variables need to be set in the `module_build_info` file. One of the `LIBRARY` or `LIBRARIES` variables must be set.

`LIBRARY` This variable specifies the name of the library to be created, for example:

```
LIBRARY = my_library
```

`LIBRARIES` This variable can be set instead of the `LIBRARY` variable to specify that several libraries should be built (with different build flags), for example:

Publication Date: 2012/3/20 Document Number: X4954A

```
LIBRARY = my_library my_library_debug
```

The first library in this list is the default library that will be linked in when an application includes this module. The application can specify one of the other libraries by adding its name to its MODULE_LIBRARIES list.

LIB_XCC_FLAGS_*libname*
: This variable can be set to the flags passed to xcc when compiling the library libname. This option can be used to pass different compilation flags to different variants of the library.

EXPORT_SOURCE_DIRS
: This variable should contain a space separated list of directories that are *not* to be compiled into the library and distributed as source instead, for example:

```
EXPORT_SOURCE_DIRS = src include
```

2 The module Makefile

Modules that build to a library can have a Makefile (unlike normal, source-only modules). The contents of this Makefile just needs to be:

```
XMOS_MAKE_PATH ?= ../..
include $(XMOS_MAKE_PATH)/xcommon/module_xcommon/build/Makefile.library
```

This Makefile has two targets. Running `make all` will build the libraries. Calling the target `make export` will create a copy of the module in a directory called `export` which does not contain the library source. For the above example, the exported module would look like the following.

```
export/
  module_my_library/
     module_build_info
     lib/
       xs1b/
         libmy_library.a
       src/
         support_fns.xc
       include/
         my_library.h
```

3 Using the module

An application can use a library module in the same way as a source module (including the module name in the USED_MODULES list). Either the module with the library source or the exported module can be used with the same end result.