**Introduction to Android.mk:**

This document describes the syntax of Android.mk build file written to describe your C and C++ source files to the Android

NDK. To understand what follows, it is assumed that you have read the docs/OVERVIEW.html file that explains their role and

usage.

An Android.mk file is written to describe your sources to the build system. More specifically:

- The file is really a tiny GNU Makefile fragment that will be parsed one or more times by the build system. As such, you

should try to minimize the variables you declare there and do not assume that anything is not defined during parsing.

- The file syntax is designed to allow you to group your sources into 'modules'. A module is one of the following:

- a static library

- a shared library

Only shared libraries will be installed/copied to your application package. Static libraries can be used to generate

shared libraries though.

You can define one or more modules in each Android.mk file, and you can use the same source file in several modules.

- The build system handles many details for you. For example, you don't need to list header files or explicit dependencies between

generated files in your Android.mk. The NDK build system will compute these automatically for you.

This also means that, when updating to newer releases of the NDK, you should be able to benefit from new toolchain/platform support

without having to touch your Android.mk files.

Note that the syntax is \*very\* close to the one used in Android.mk files distributed with the full open-source Android platform sources. While

the build system implementation that uses them is different, this is an intentional design decision made to allow reuse of 'external' libraries'

source code easier for application developers.