**NDK-provided variables:**

These GNU Make variables are defined by the build system before your Android.mk file is parsed. Note that under certain circumstances

the NDK might parse your Android.mk several times, each with different definition for some of these variables.

CLEAR\_VARS

Points to a build script that undefines nearly all LOCAL\_XXX variables listed in the "Module-description" section below. You must include

the script before starting a new module, e.g.:

include $(CLEAR\_VARS)

BUILD\_SHARED\_LIBRARY

Points to a build script that collects all the information about the module you provided in LOCAL\_XXX variables and determines how to build

a target shared library from the sources you listed. Note that you must have LOCAL\_MODULE and LOCAL\_SRC\_FILES defined, at a minimum before

including this file. Example usage:

include $(BUILD\_SHARED\_LIBRARY)

note that this will generate a file named lib$(LOCAL\_MODULE).so

BUILD\_STATIC\_LIBRARY

A variant of BUILD\_SHARED\_LIBRARY that is used to build a target static library instead. Static libraries are not copied into your

project/packages but can be used to build shared libraries (see LOCAL\_STATIC\_LIBRARIES and LOCAL\_WHOLE\_STATIC\_LIBRARIES described below).

Example usage:

include $(BUILD\_STATIC\_LIBRARY)

Note that this will generate a file named lib$(LOCAL\_MODULE).a

PREBUILT\_SHARED\_LIBRARY

Points to a build script used to specify a prebuilt shared library.

Unlike BUILD\_SHARED\_LIBRARY and BUILD\_STATIC\_LIBRARY, the value of LOCAL\_SRC\_FILES must be a single path to a prebuilt shared

library (e.g. foo/libfoo.so), instead of a source file.

You can reference the prebuilt library in another module using the LOCAL\_PREBUILTS variable (see docs/PREBUILTS.html for more

information).

PREBUILT\_STATIC\_LIBRARY

This is the same as PREBUILT\_SHARED\_LIBRARY, but for a static library file instead. See docs/PREBUILTS.html for more.

TARGET\_ARCH

Name of the target CPU architecture as it is specified by the full Android open-source build. This is 'arm' for any ARM-compatible

build, independent of the CPU architecture revision.

TARGET\_PLATFORM

Name of the target Android platform when this Android.mk is parsed.

For example, 'android-3' correspond to Android 1.5 system images. For a complete list of platform names and corresponding Android system

images, read docs/STABLE-APIS.html.

TARGET\_ARCH\_ABI

Name of the target CPU+ABI when this Android.mk is parsed.

Two values are supported at the moment:

armeabi For ARMv5TE

armeabi-v7a

NOTE: Up to Android NDK 1.6\_r1, this variable was simply defined as 'arm'. However, the value has been redefined to better

match what is used internally by the Android platform.

For more details about architecture ABIs and corresponding compatibility issues, please read docs/CPU-ARCH-ABIS.html

Other target ABIs will be introduced in future releases of the NDK and will have a different name. Note that all ARM-based ABIs will

have 'TARGET\_ARCH' defined to 'arm', but may have different

'TARGET\_ARCH\_ABI'

TARGET\_ABI

The concatenation of target platform and ABI, it really is defined as $(TARGET\_PLATFORM)-$(TARGET\_ARCH\_ABI) and is useful when you want

to test against a specific target system image for a real device.

By default, this will be 'android-3-armeabi'

(Up to Android NDK 1.6\_r1, this used to be 'android-3-arm' by default)