

# A safari walk-through into JNI within Android™ OS

Exploiting JNI capabilities taking advantage of Android™ NDK

Antonio Troina

Matr. 708267, (antonio.troina@mail.polimi.it)

*Introductory report for the M.Sc. thesis in Computer Science Engineering*

*Reviser: PhD. Patrick Bellasi (bellasi@elet.polimi.it)*

Last update: October 28, 2012

## Abstract

This article aims to briefly describe, in the form of some simple and annotated tutorials, how a developer can take advantage of some JNI capabilities under Android operating system to let the Java and Native environments communicate to each other. This operation has been made quite easy by the Android's native development kit (NDK). Special attention will be given to the callback mechanism, which is by far the most complex, nevertheless the most challenging, under the developer's point of view.

## 1 Introduction

The coming of Android in the smart-phones market would suggest that Java is definitely enough to rule this galaxy. Moreover, lately, the little-green-robot's operative system has approached, silently but firmly, to the embedded world which, typically, wasn't famous for being dominated by the Java language, particularly for its performance-oriented needs. This being said, except for the peculiar case of the *JIT-ed* code, who's responsible of connecting the high-level Java application layer to the native world, and the other way around? Yes, this interface is JNI, which was initially released in early 1997. So far, some basic examples are available (mainly on-line) which, however, are often more theoretical than practical, therefore this article is meant to be a concrete hands-on guide, to discover - some of - the secrets behind this powerful instrument which is JNI.

### 1.1 JNI, Android and NDK

JNI per se basically needs two components to be used: the *javah* JDK tool, which builds c-style header files from a given Java class (that will be implemented afterwards in a proper native source file, which includes the mentioned header), and the *jni.h* header file, which maps the Java types to their native counterparts.

## 2 Conclusions

This report and all the source code are publicly available through the git repository at <https://github.com/thoeni/ndk-tutorials> or at <https://bitbucket.org/atroina/ndk-tutorials>.

## References

- [1] Liang, S.: The Java Native Interface. Addison-Wesley (1999)
- [2] Gargenta, A.: Jni reference example (Oct. 2012)