Open NFC - Android ICS 4.0.4 - Porting Guide

Document Type:	Manual
Reference:	MAN_NFC_1205-326 Version 0.1 (14114)
Release Date:	May 16, 2012
File Name:	MAN_NFC_1205-326 Open NFC - Android ICS 4.0.4 - Porting Guide v0.1.pdf
Security Level:	General Business Use

Disclaimer

History

Version	Date	Comments
0.1	May 16, 2012	First Release

Summary of Contents

1 Introduction

This document is the quick version of Android porting guide for the NFC software stack "OpenNFC 4.4.2".

This document describes the porting of OpenNFC4.4.2 for Android ICS 4.0.4 or an extended version of ICS 4.0.4.

2 Building Instructions

The first part refers to retrieving Android Open Source ICS4.0.4. If you have already done or have an extended version of ICS4.0.4, please go to section 2.6.

2.1 Create bin/ directory and include it in path

```
$ mkdir ~/bin
$ PATH=~/bin:$PATH
```

2.2 Download the Repo script and ensure it is executable

```
$ curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

2.3 Make a New working directory (i.e. android-4.0.4_r1)

```
mkdir android-4.0.4_r1 && cd android-4.0.4_r1
```

2.4 Run repo init to bring down the latest version of Repo

```
$ repo init -u https://android.googlesource.com/platform/manifest -b android-4.0.4_r1
```

2.5 Run repo sync to pull down files

```
$ repo sync
```

2.6 Getting last Open NFC sources

Retrieve and unzip the archive package:
Delivery_ICS4.0.4_OpenNFC4.4.2_2012_05_25_207cd863574c127df61c6eecc3f6e0f8.tar.gz

```
$ tar zxvf  Delivery_ICS4.0.4_OpenNFC4.4.2_2012_05_25_207cd863574c127df61c6eecc3f6e0f8.tar.gz
```

You will find the following hierarchy of the unzipped files:
(Hints: you can use "tree -d PATH_OF_UNZIPPED_FILE")

```
Hierarchy
├── Demo - ChangePolicy
├── Kernel
│   ├── kernel for AOSP
│   │   └── nfc
│   └── kernel_for_NFCRealDevice
│       └── nfc
├── libnfc-opennfc
│   ├── java
│   │   ├── jni
│   │   └── src
│   │       └── com
│   │           └── opennfc
│   │               └── extension
│   │                   └── nfc
│   │                       └── api
│   ├── ndef
│   ├── open_nfc
│   │   ├── hardware
│   │   │   └── libhardware
│   │   │       ├── include
│   │   │       │   └── hardware
```

```
          └── modules
              └── nfcc
                  ├── nfcc
                  ├── nfc_hal_microread
                  │   ├── interfaces
                  │   ├── porting
                  │   └── sources
                  └── nfc_hal_simulator
                      ├── interfaces
                      ├── porting
                      └── sources
      ├── open nfc
      │   ├── interfaces
      │   ├── porting
      │   │   ├── client
      │   │   │   └── jni
      │   │   │       └── headers
      │   │   ├── common
      │   │   └── server
      │   │       └── jni
      │   └── sources
      └── standalone server
          └── src
  ├── open_nfc_extension
  ├── patches
  └── test
└── Nfc
    └── jni
```
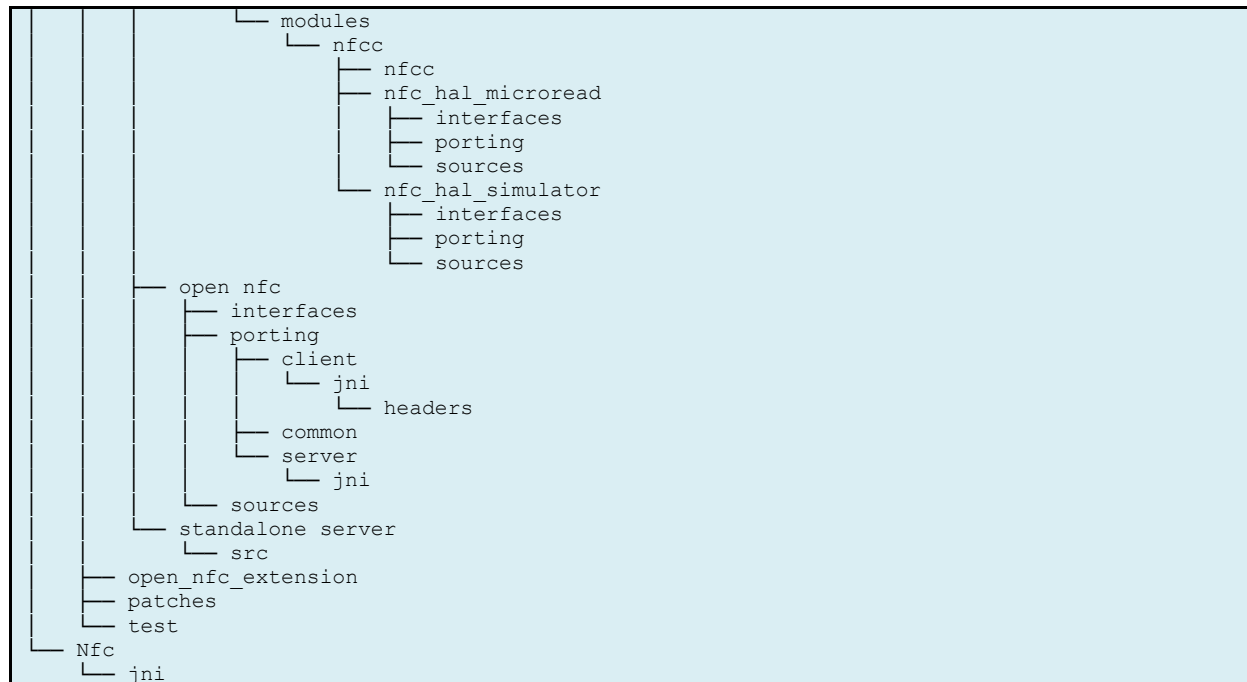
2.7 Modify Android source code

The following section explains which modifications should be done for the Android source code.

If you use the android official source code from AOSP ICS-4.0.4, you can apply directly the patches included in the Open NFC package. All details are explained in the section "Modifications of the Android source code by patches".

If you use a modified or extended Android source code ICS-4.0.4, you should change it manually in accordance with the instructions provided below in the section "Manual modifications of the Android source code".

Thus a user should follow only one set of instructions: either Modifications of the Android source code by patches or Manual modifications of the Android source code.

2.7.1 Modifications of the Android source code by patches

2.7.1.1 Copy libnfc-opennfc folder

Copy the unzipped subfolder libnfc-opennfc under the repository
<ANDROID_ROOT>/external/

You will find all the patches under <ANDROID_ROOT>/external/libnfc-opennfc/patches. These patches can be used to modify the original Android source code only:
- build.patch
- frameworks_base.patch
- system_core.patch
- external_libnfc-nxp.patch
- packages_apps_Nfc.patch
- kernel.patch (used later in section 2.9.3)

export the ANDROID_BUILD_TOP

```
cd <ANDROID ROOT DIR>
export ANDROID_BUILD_TOP=$PWD
```

You can either **apply all the patches by launching apply_patches.sh (in this case, go to section 2.7.1.6 after launching apply_patches.sh)** or **apply these patches one by one (instructions in section 2.7.1.1 - 2.7.1.5)**

2.7.1.2 Apply build.patch

```
android-4.0.4_r1$ cd android-4.0.4_r1/build
android-4.0.4_r1/build$ git apply --whitespace=nowarn PATH_OF_build.patch
```

2.7.1.3 Apply system_core.patch

```
android-4.0.4 r1/build$ cd ../system/core
android-4.0.4_r1/system/core$ git apply --whitespace=nowarn PATH_OF_system_core.patch
```

2.7.1.4 Apply the patch frameworks_base.patch

```
android-4.0.4 r1/system/core$ cd ../../frameworks/base
android-4.0.4_r1/frameworks/base$ git apply --whitespace=nowarn PATH_OF_frameworks_base.patch
```

2.7.1.5 Apply the patch external_libnfc-nxp.patch

```
android-4.0.4_r1/frameworks/base$ cd ../../external/libnfc-nxp
android-4.0.4_r1/external/libnfc-nxp$ git apply --whitespace=nowarn PATH_OF_external_libnfc-nxp.patch
```

2.7.1.6 Apply the patch packages_apps_Nfc.patch

```
android-4.0.4_r1/external/libnfc-nxp$ cd ../../packages/apps/Nfc
android-4.0.4 r1/packages/apps/Nfc$ git apply --whitespace=nowarn
PATH_OF_packages_apps_Nfc.patch
```

2.7.1.7 Copy Nfc/jni folder

Remove the original folder <ANDROID_ROOT>/packages/apps/Nfc/jni, and Copy the unzipped subfolder "Nfc/jni" under the repository <ANDROID_ROOT>/packages/apps/Nfc/.

2.7.2 Manual modifications of the Android source code

If a developer uses an already modified or extended Android source code, he should change it manually in accordance with the instructions provided below.

2.7.2.1 Modifications for <ANDROID_ROOT>/build/core/base_rules.mk

The following error message should be changed to a warning:

```
ifneq ($(strip
$(LOCAL_BUILT_MODULE)$(LOCAL_INSTALLED_MODU
LE)),)
  $(error $(LOCAL_PATH): LOCAL_BUILT_MODULE
and LOCAL_INSTALLED_MODULE must not be
defined by component makefiles)
endif
```

=>

```
ifneq ($(strip
$(LOCAL_BUILT_MODULE)$(LOCAL_INSTALLED_MODULE
)),)
  $(warning $(LOCAL_PATH): LOCAL_BUILT_MODULE
and LOCAL_INSTALLED_MODULE must not be
defined by component makefiles)
endif
```

2.7.2.2 Modifications for <ANDROID_ROOT>/build/core/main.mk

The following code should be commented out:

```
ifeq ($(stash product vars),true)
$(call assert-product-vars,  STASHED)
endif
```

=>

```
ifeq ($(stash product vars),true)
# $(call assert-product-vars,  STASHED)
endif
```

2.7.2.3 Modifications for <ANDROID_ROOT>/frameworks/base/data/etc/platform.xml

The following lines should be added inside the <permissions> </permissions> block:

```
    <assign-permission
name="android.permission.ACCESS_SURFACE_FLI
NGER" uid="graphics" />


    <!-- This is a list of all the
libraries available for application
        code to link against. -->
```

=>

```
    <assign-permission
name="android.permission.ACCESS_SURFACE_FLING
ER" uid="graphics" />

    <!-- Feature NFC  -->
    <feature name="android.hardware.nfc" />


    <!-- This is a list of all the libraries
available for application
        code to link against. -->
```

2.7.2.4 Modifications for <ANDROID_ROOT>/system/core/rootdir/init.rc

The following lines should be added:

```
service debuggerd /system/bin/debuggerd
    class main
```

=>

```
service debuggerd /system/bin/debuggerd
    class main

#Open NFC server
service server open nfc
/system/bin/server_open_nfc
    class main
    user root
    group root
```

Important: There are **4 spaces** before *class, user and group*. It is really important to respect this rule.
An empty line **MUST** be inserted between two services descriptions.

2.7.2.5 Modifications for <ANDROID_ROOT>/system/core/rootdir/ueventd.rc

The following lines should be added:

```
# sysfs properties
/sys/devices/virtual/input/input*   enable
0660  root   input
```

=>

```
#NFC device
/dev/nfcc                       0600   root
root

# sysfs properties
/sys/devices/virtual/input/input*   enable
0660  root   input
```

2.7.2.6 Modifications for <ANDROID_ROOT>/external/libnfc-nxp

In order to avoid the conflict with original JNI implementation, remove "**Android.mk**" and "**CleanSpec.mk**" files from this directory

2.7.2.7 Modifications for <ANDROID_ROOT>/packages/apps/Nfc

The following changes should be done manually.

a) <ANDROID_ROOT>/packages/apps/Nfc/Android.mk
The NFC code should be compiled. The "eng" tag is added to the **LOCAL_MODULE_TAGS** to compile the NFC-related code**.**

`LOCAL_MODULE_TAGS := optional`	**=>**	`LOCAL_MODULE_TAGS := optional eng`

b) <ANDROID_ROOT>/packages/apps/Nfc/src/com/android/nfc/SendUi.java
This modification is required only for Android Emulator, emulator taking screenshot fails during the Beam animation and the P2P can't be finished. To avoid this problem, if the method returns null, "a fake screenshot" is created.

```Bitmap bitmap = Surface.screenshot((int) dims[0], (int) dims[1]); // Bail if we couldn't take the screenshot if (bitmap == null) { return null; }```	**=>**	```Bitmap bitmap = Surface.screenshot((int) dims[0], (int) dims[1]); // Bail if we couldn't take the screenshot if (bitmap == null) { int l = (int) dims[0] * (int) dims[1]; int[] pixels = new int[l]; for (int i = 0; i < l; i++) pixels[i] = 0xFF000000	((int) (Math.random() * 0XFF)) << 16	((int) (Math.random() * 0XFF)) << 8	((int) (Math.random() * 0XFF)); bitmap = Bitmap.createBitmap(pixels, (int)dims[0], (int) dims[1], Bitmap.Config.ARGB_8888); }```

c) <ANDROID_ROOT>/packages/apps/Nfc/src/com/android/nfc/nxp/NativeNfcManager.java
This modification is to change the firmware binary file's name.

```public class NativeNfcManager implements DeviceHost { private static final String TAG = "NativeNfcManager"; private static final String NFC CONTROLLER FIRMWARE FILE NAME = "/vendor/firmware/libpn544_fw.so"; private static final String PREF = "NxpDeviceHost";```	**=>**	```public class NativeNfcManager implements DeviceHost { private static final String TAG = "NativeNfcManager"; private static final String NFC CONTROLLER FIRMWARE FILE NAME = "/vendor/firmware/microread fw.bin"; private static final String PREF = "NxpDeviceHost";```

2.7.2.8 Copy libnfc-opennfc folder

Copy the unzipped repository libnfc-opennfc under <ANDROID_ROOT>/external/

2.7.2.9 Copy Nfc/jni folder

Remove the original folder <ANDROID_ROOT>/packages/apps/Nfc/jni, and Copy the unzipped subfolder "Nfc/jni" under the repository <ANDROID_ROOT>/packages/apps/Nfc/.

2.8 Configuration for compilation

```
cd (ANDROID_ROOT_DIR)
export PATH=$PATH:$PWD/prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin
export ANDROID_BUILD_TOP=$PWD
export CROSS_COMPILE=arm-eabi- ARCH=arm
```

2.9 Kernel Compilation

If you have already a specific kernel, go to section 2.9.3.

2.9.1 Download the source code of the linux kernel

```
mkdir kernel
cd kernel
git clone https://android.googlesource.com/kernel/goldfish.git .
git checkout -b android-goldfish-2.6.29
git pull origin android-goldfish-2.6.29
```

2.9.2 Generate the default android emulator Linux kernel:

```
cd (ANDROID_ROOT_DIR)/kernel
make goldfish_armv7_defconfig
```

2.9.3 Modify the kernel code

The folder "Unzipped_Folder/Kernel/kernel_for_NFCRealDevice/nfc" is for specific kernel of OEM and a real Android NFC device/testing board.
The folder "Unzipped_Folder/Kernel/kernel_for_AOSP" is for original kernel and Android Emulator (this folder can also be automatically generated by applying the kernel.patch in section 2.9.3.1).

2.9.3.1 In case of the original kernel code, a user can apply directly the provided kernel patch

```
git apply (ANDROID_ROOT_DIR)/external/libnfc-opennfc/patches/kernel.patch
```

2.9.3.2 In case of OEM's specific kernel, a user should do the following modifications manually

a) **kernel/Makefile (or according your specific kernel path)**

drivers-y := drivers/ sound/ firmware/	**=>**	drivers-y := drivers/ sound/ firmware/ nfc/

b) kernel/arch/arm/Kconfig (or according your specific kernel path)

source "lib/Kconfig"	**=>**	source "lib/Kconfig" source "nfc/Kconfig"

IMPORTANT! The empty line should be added between these two source lines.

c) Copy the subfolder "Unzipped_Folder/Kernel/kernel_for_NFCRealDevice/nfc" under "<ANDROID_ROOT>/kernel/" (or under your own specific kernel repository)

Modify the NFC-related files for a real NFC device

The next two points are brief description of files inside the "<ANDROID_ROOT>/kernel/nfc" folder and the instructions for modifications of these files for a real Android NFC device/testing board.

d) Makefile

This is a Makefile for compilation of NFC-related source files by the kernel.

The following lines are required **only for emulator version** for communication with the Connection Center, therefore make sure the following lines are not included in the Makefile.

```
microread-objs += ccclient.o
microread-objs += ccclient_md.o
```

If OpenNFC is implemented in **a real NFC device.** Make sure the following files are not included under the repository <ANDROID_ROOT>/kernel/nfc.
ccclient.h, ccclient.c, ccclient_md.h, ccclient_md.c

e) open_nfc_custom.c

This file must be changed if OpenNFC is implemented in **a real NFC device.**
After doing the above step (c), you will find a template file **open_nfc_custom.c** under "<ANDROID_ROOT>/kernel/nfc". This file must be modified according to the real Android NFC device/testing board.

The main adaptation consists in :
    - definition of the 2 GPIOs (IRQOUT, and RST/WakeUP), and their initialization.
    - the name of the I2C driver available for MicroRead ("MicroRead" in this example).

Typically **4 lines** should be modified according to the real Android NFC device/testing board:
    - Platform IRQ Number used with IRQOUT, if not dynamically allocated
    - GPIO_NFC_IRQOUT definition (IRQOUT pin of MicroRead)
    - GPIO_NFC_WAKEUP definition (RESETWAKE_UP pin of MicroRead)
    - "struct i2c_device_id" line for I2C Slave definition for MicroRead and the initialization of the I2C bus (i2c_get_adapter, ...).

2.9.4 Configure make for kernel compilation

Go to Android kernel path or your specific kernel path:
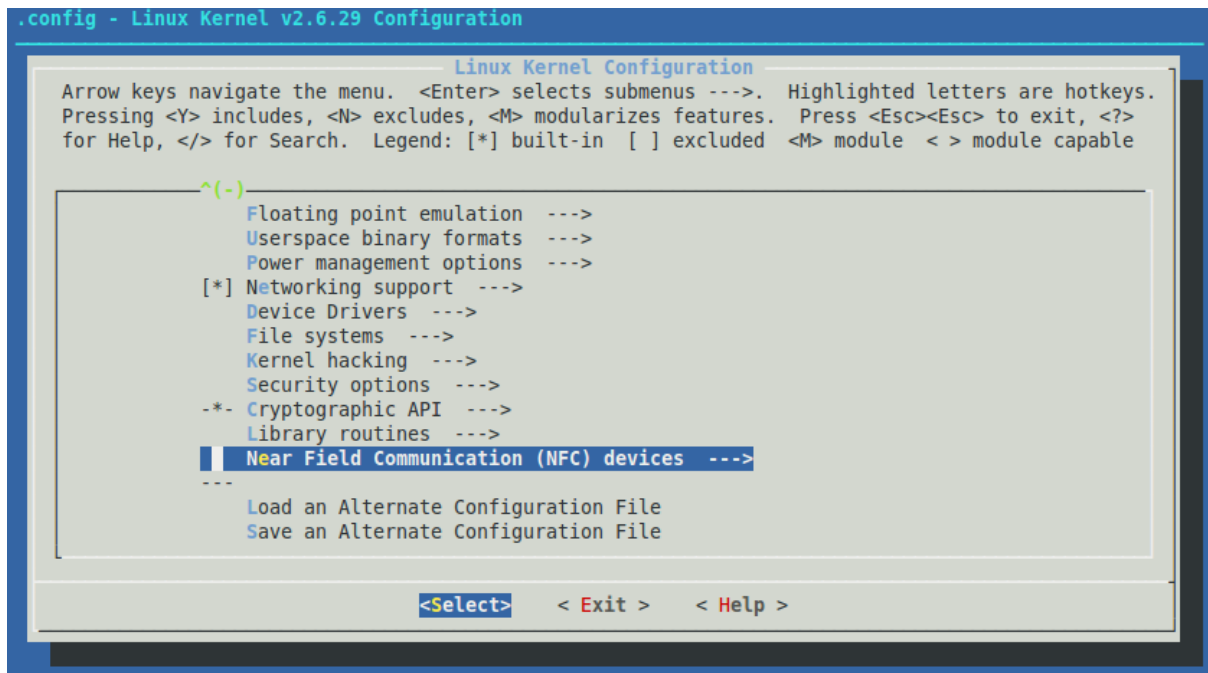
```
cd (ANDROID_ROOT_DIR)/kernel
make menuconfig
```

```
 .config - Linux Kernel v2.6.29 Configuration

                           Linux Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

       ^(-)
              Floating point emulation  --->
              Userspace binary formats  --->
              Power management options  --->
        [*] Networking support  --->
              Device Drivers  --->
              File systems  --->
              Kernel hacking  --->
              Security options  --->
        -*- Cryptographic API  --->
              Library routines  --->
              Near Field Communication (NFC) devices  --->
        ---
              Load an Alternate Configuration File
              Save an Alternate Configuration File


                <Select>    < Exit >    < Help >
```

Select "Near Field Communication (NFC) devices", press <Enter>.

```
 .config - Linux Kernel v2.6.29 Configuration

                        Near Field Communication (NFC) devices
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable


        [*] Near Field Communication (NFC) devices
              Choose NFC implementation (MICROREAD driver)  --->

                <Select>    < Exit >    < Help >
```

Select "Near Field Communication (NFC) devices" and press <SPACE BAR>.
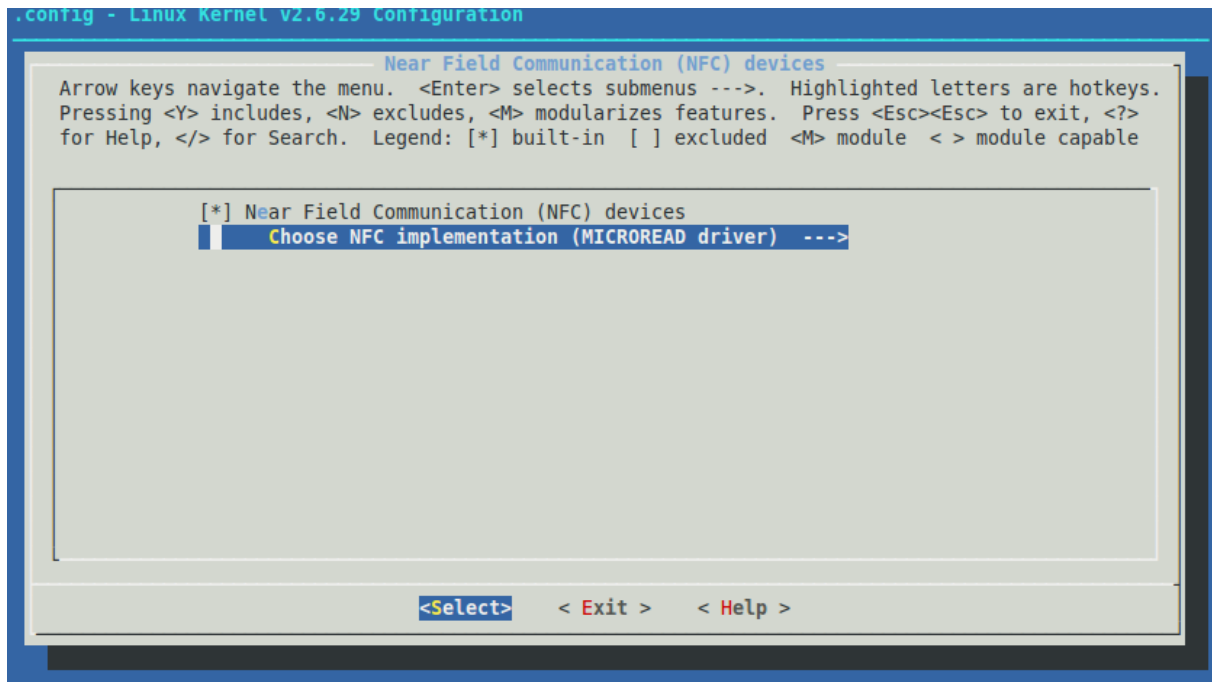
Select "Choose NFC implementation (MICROREAD driver)", and press <Enter>.
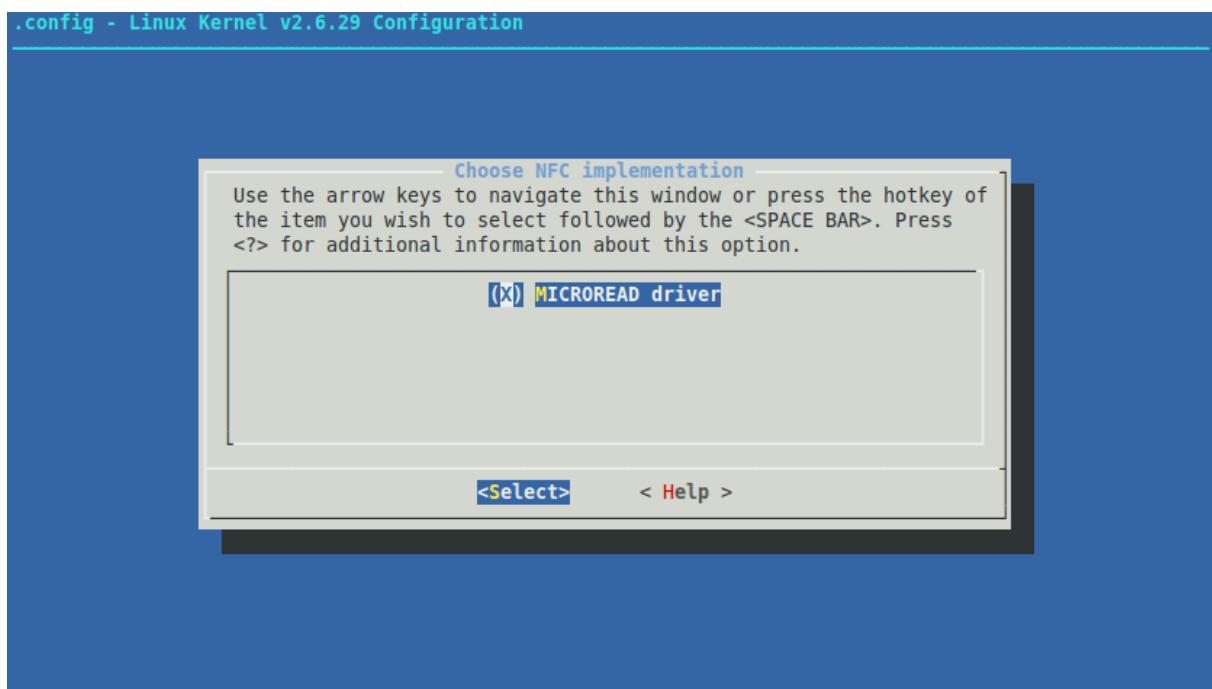
Press "ENTER" to select.

Return back to the main menu via "Exit" (twice) and save the parameters setting.

2.9.5 Compile the modified kernel

Recommendation: enable CCACHE and export the following variable USE_CCACHE and enhance the ccache size following your available space disk.

```
$ export USE_CCACHE=1
$ ./prebuilt/linux-x86/ccache/ccache –M15G
```

Then compile kernel
```
$ make –j2
```

2.10 Final Compilation

Once Open NFC Android successfully installed in the standard AOSP sources, the lunch command should be run in the AOSP tree:

```
$ . ./build/envsetup.sh

$ lunch

You're building on Linux

Lunch menu... pick a combo:
    1. full-eng
    2. full_x86-eng
    3. vbox_x86-eng
    4. full_maguro-userdebug
    5. full_tuna-userdebug
    6. full_panda-eng
Which would you like? [full-eng] 1    (for emulator)

$ make –j4
```

2.11 Configuration the IP address for Connection Center in case of using Android Emulator

If you test on an android emulator via connection center, please do the following 5 steps:
   a) Wait until the compilation is done.
   b) Configure the file "<ANDROID_ROOT>/external/libnfc-opennfc/connection_center_access" within the following format
      <Type> <IP>
      <Type>: "M" for Microread and "S" for Simulator
      <IP>: IP address of the machine where the "Connection Center" is running.
      Note: please leave exactly one space between the two elements

   c) Launch the emulator
   d) Push the configured file into "/data" of emulator with the command:

```
adb push < ANDROID_ROOT >/external/libnfc-
opennfc/connection_center_access /data
```
   e) Close and Re-launch the emulator

Note: When OpenNFC architecture is run on the Android Emulator via Connection Center, it can take much longer for reading/writing/P2P operations compared to the cases where a real NFC device is involved.

3 License

The source files of the driver sample of the "Open NFC for Linux" are distributed using the "Apache v2.0" license. They can be freely adapted or modified and used to create a dynamic driver module or to be statically linked with the kernel without any license issue.

The remaining of the Open NFC core source code and the source code of the "Open NFC for Android" are distributed under Apache v2.0 license.