

Rebuilding the NDK Core

Contents

Overview

Prerequisites

Preparing the Build Directory

Selecting Software Versions in CCSv4

Performing the Re-Build

Building Your Project Using the Rebuilt NDK Core Libraries for DSP/BIOS 5 Users

Building Your Project Using Rebuilt NDK Core Libraries for SYS/BIOS 6 Users

Note: These instructions apply to NDK 2.20. To rebuild libraries for NDK 2.21 using Gmake, please refer to http://processors.wiki.ti.com/index.php/Rebuilding_The_NDK_Core_Using_Gmake

Overview

The Network Developer's Kit (NDK) core includes source files and a CCSv4 project that allow you to modify the stack sources and rebuild its libraries. You can rebuild the NDK core in order to modify, update, or add functionality. If you edit the NDK source code, an NDK project file, and/or an NDK RTSC build script, you must also rebuild the NDK core in order to create new libraries containing these modifications.

The instructions on this page are intended for rebuilding the NDK on Microsoft Windows using CCSv4. However, the NDK core libraries may also be re-built on Linux using the xdc command that comes packaged with XDCtools.

Warning: This appendix provides details about rebuilding the NDK stack source code. We strongly recommend that you copy the NDK installation to a directory with a different name and rebuild that copy, rather than rebuilding the original installation.

Prerequisites

In order to rebuild the NDK core libraries, you must have already installed the following products. If you have installed CCSv4, both SYS/BIOS and XDCtools are installed as part of that installation.

- NDK
- SYS/BIOS 6.x or DSP/BIOS 5.x
- XDCtools

It is important to build the NDK with compatible versions of SYS/BIOS (or DSP/BIOS) and XDCtools. To find out which versions are compatible, see the "Validation" section of the *Release Notes* in the top-level directory of your NDK installation. You must build with the versions listed in the "Validation" section or later versions.

Prior to importing the NDK core build project, you must select the proper versions of SYS/BIOS. See [Selecting Software Versions in CCSv4](#) for details on how to do this.

Preparing the Build Directory

Follow these steps to prepare to rebuild the NDK source code:

1. Create a new directory on your computer. You will use this directory to store a copy of the NDK installation. This directory will act as a container for your NDK modifications. The full path to this directory *cannot contain any spaces*. For example, we recommend creating a directory called "C:\myNdkBuilds" rather than using a location in the "My Documents" directory tree.
2. Using Windows Explorer, copy the entire NDK installation to the directory you just created. For example, if you installed the NDK in the default installation location, then you should copy the C:\Program Files\Texas Instruments\ndk_2_20_### folder and all of its contents to the C:\myNdkBuilds folder. After this step, the C:\myNdkBuilds folder should contain the ndk_2_20_### folder, which is a copy of your NDK installation.
3. Rename the folder that contains the copy of the NDK installation. For example, rename the C:\myNdkBuilds\ndk_2_20_### directory to C:\myNdkBuilds\custom_ndk_2_20_###.
4. Using Windows Explorer, move to the C:\myNdkBuilds\custom_ndk_2_20_###\packages\ti\ndk directory.
5. In this directory, find the file called "config.bld.default". Rename this file to be "config.bld".
6. Open the config.bld file for text editing.
7. Make sure that the paths to the SYS/BIOS (or DSP/BIOS) and Code Generation Tools are correct. For example, if you have installed SYS/BIOS 6.30 and wish to use this version with the default Code Gen Tools that came with your CCSv4 installation, set the variables "bios6Path" and "rootDir" as follows:

```
var bios6path = "D:/Program Files/Texas Instruments/bios_6_30_00_28/packages";  
  
// location of Code Generation Tools  
var rootDir = "D:/Program Files/Texas Instruments/ccsv4/tools/compiler/c6000"
```

8. Near the end of the file, look for the array called "Build.targets". This array contains the list of targets for which the NDK should be built. Make sure that the target for which you want the NDK built is uncommented. For example, if you want to build the NDK for the C64P target only, then your Build.targets array might look like this:

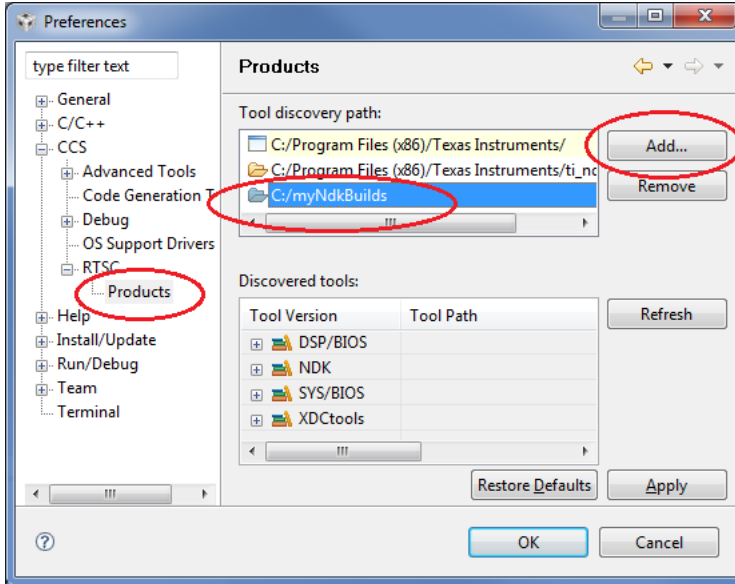
```
Build.targets = [  
    tiTargets.C64P,  
    // tiTargets.C64P_big_endian,  
    // tiTargets.C674,  
    // tiTargets.C674_big_endian,  
    // armElfTargets.Arm9,  
    // armElfTargets.A8F,  
];
```

9. Save and exit the file.

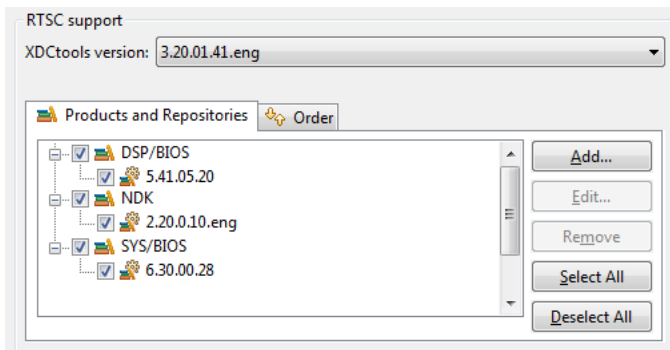
Selecting Software Versions in CCSv4

In CCSv4, you must make sure that the proper versions of XDCtools, BIOS and NDK are selected before rebuilding the NDK. Do this by editing the CCSv4 global preferences as follows:

1. In CCSv4, choose the **Window > Preferences** menu command.
2. Go to the **CCS > RTSC > Products** category.
3. Next to the "Tool discovery path" click the **Add** button.
4. Navigate to the location of your custom copy of the NDK installation. For example, navigate to `C:\myNdkBuilds`.
5. Click **OK** and you should see this path appear:



6. Go to the **CCS > RTSC** category, and make sure that the proper versions of XDCtools, BIOS, and NDK are selected:

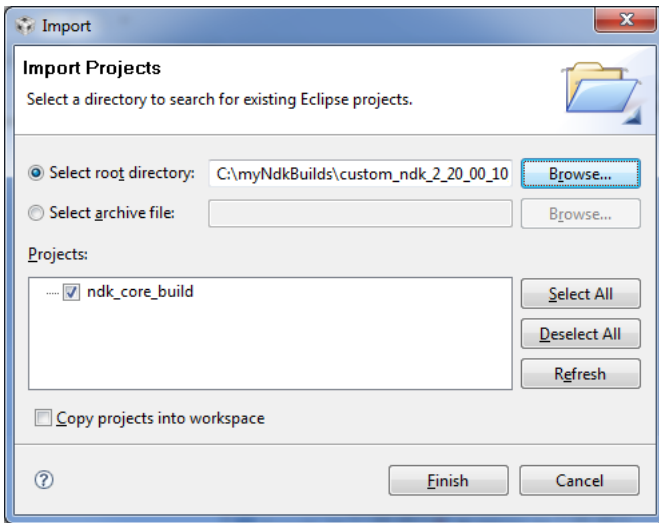


7. Click **OK** to close the CCSv4 Preferences window.

Performing the Re-Build

Follow these steps to rebuild the NDK source code:

1. Make sure that you are in the C/C++ perspective of CCSv4.
2. Select the **Project > Import Existing CCS/CCE Eclipse Project** menu command.
3. In the window that appears, choose "Select root directory", then click the **Browse** button.
4. Find the location of your copy of the NDK. For example `C:\myNdkBuilds\custom_ndk_2_20_###\packages\ti\ndk`.
5. With the custom NDK directory selected, click **OK**. You should see a project titled "ndk_core_build" in the list of projects.
6. Check the box next to "ndk_core_build" and make sure that there is no check next to "Copy projects into workspace" as shown in the next figure:



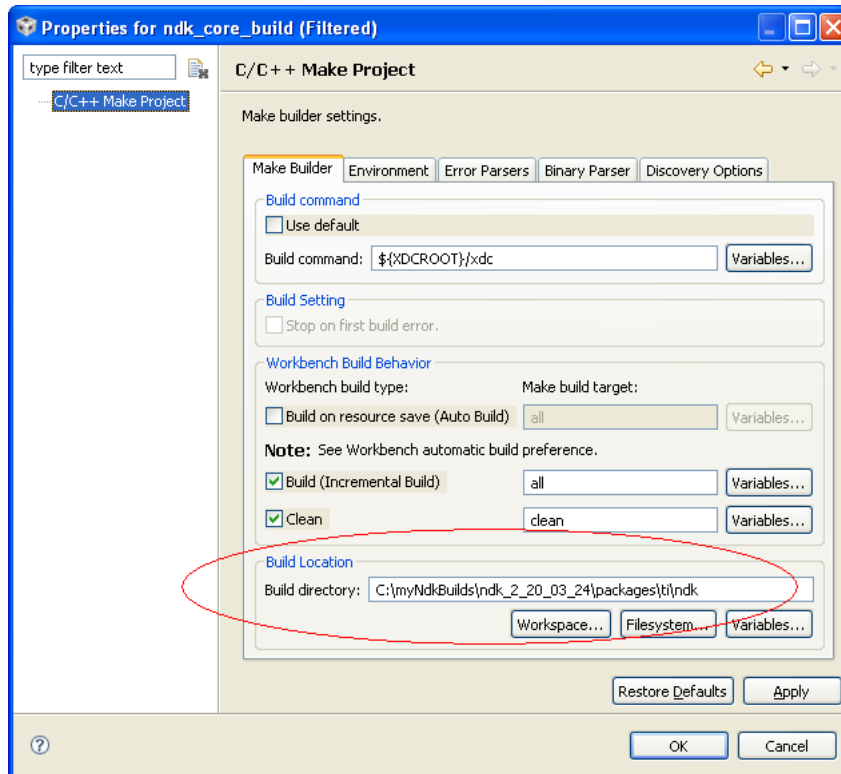
7. Click **Finish** to import the project.

- **Note:** If you already have a project named "ndk_core_build" in your current workspace, then the project will not be imported. If this is the case, you should create a new workspace and repeat the steps in this section.
- **Note:** There is currently a bug which causes the NDK environment variable specified in the ndk_core_build project to not be resolved properly. This bug causes an error similar to the following when rebuilding the project:

Error launching builder (C:/Program Files/Texas Instruments/xdctools_3_21_00_27_eng/xdc all)

(Exec error:The directory name is invalid.)

To work around this issue, you should edit the project build options to explicitly specify the path to the copy of the NDK installation. This is illustrated in following screen shot:



8. Click **OK** to save the project build options.

9. Rebuild the project by selecting **Project > Rebuild All**.

Building Your Project Using the Rebuilt NDK Core Libraries for DSP/BIOS 5 Users

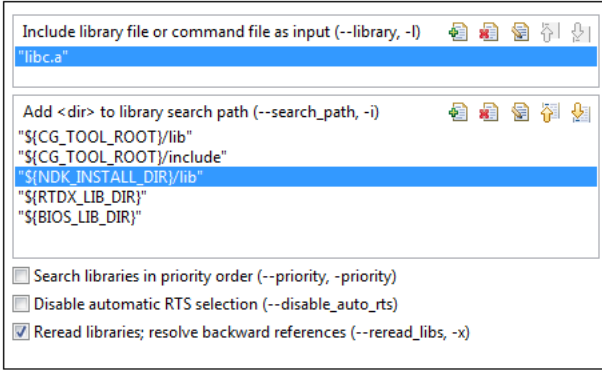
To build your DSP/BIOS 5 application using the version of NDK you have rebuilt, you must update the environment variable %NDK_INSTALL_DIR% to point to the rebuilt version by following these steps:

1. In your computer's system settings, update the NDK_INSTALL_DIR environment variable to point to the location of your rebuilt NDK. For example, set it to:

```
NDK_INSTALL_DIR = C:\myNdkBuilds\custom_ndk_2_20_##-##
```

2. Open CCSv4 and select the application project you want to rebuild. If CCSv4 was already open when you set the environment variable, you must close and restart CCSv4 in order for the environment variable change to take effect.
3. Right-click on your project and choose **Build Properties**.

- In the **C/C++ Build** category of the Properties dialog, under **C6000 Linker**, choose **File Search Path**.
- If your project already references the NDK core libraries using the NDK_INSTALL_DIR environment variable, then you are finished and may rebuild your project using the re-built version of the NDK.
- If your project does not reference the NDK core libraries using the NDK_INSTALL_DIR environment variable, update the path to the NDK libraries so that they point to the versions you rebuilt in C:\myNdkBuilds\custom_ndk_2_20_##-##.

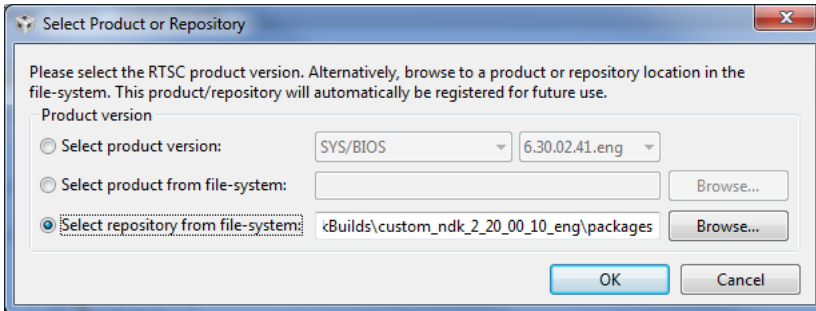


- Once you have updated all paths, click **OK**.
- Rebuild your application.

Building Your Project Using Rebuilt NDK Core Libraries for SYS/BIOS 6 Users

To build your SYS/BIOS 6 application using the version of NDK you have rebuilt, you must point your project to this rebuilt version by following these steps:

- Open CCSv4 and select the application project you want to rebuild.
- Right-click on your project and choose **Build Properties**. If you have a RTSC configuration project that is separate from your application project, open the build properties for the configuration project.
- In the **CCS Build** category of the Properties dialog, choose the **RTSC** tab.
- Under the **Products and Repositories** tab, uncheck all the boxes for NDK. This ensures that no version is selected.
- Click the **Add** button next to the Products and Repositories tab.
- Choose **Select repository from file-system**, and browse to the "packages" directory of the location where you copied and rebuilt the NDK. For example, the location may be C:\myNdkBuilds\custom_ndk_2_20_##-##\packages.



- Click **OK** to apply these changes to the project.
- You may now rebuild your project using the re-built version of the NDK.

<pre> {{ 1. switchcategory:MultiCore= ■ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum ■ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article Rebuilding the NDK Core here. </pre>	<p>Keystone=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article Rebuilding the NDK Core here.</p>	<p>C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>DaVinci=For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article Rebuilding the NDK Core here.</p>	<p>For technical support please post your questions at http://e2e.ti.com. Please post on comments about article Rebuild NDK Core here.</p>
--	--	--	---	---	--	---	--	--

Links



Amplifiers & Linear
Audio
Broadband RF/IF & Digital Radio

DLP & MEMS
High-Reliability
Interface

Processors

- ARM Processors
- Digital Signal Processors (DSP)

Switches & Multiplexers
Temperature Sensors & Control ICs
Wireless Connectivity

[Clocks & Timers](#)
[Data Converters](#)

[Logic](#)
[Power Management](#)

- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=Rebuilding_the_NDK_Core&oldid=105679"

This page was last edited on 24 April 2012, at 19:11.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.