



Technology for Innovators™

NDK 1.94.01.001 Release Notes

This release note is divided into the following sections:

[Introduction](#), [Documentation](#), [What's New](#), [Upgrading And Compatibility Information](#), [Software Installation Steps](#), [Software Installation Overview](#), [Important First Steps](#), [Host Support](#), [Dependencies](#), [Device Support](#), [Validation Information](#), [Known Issues](#), [Benchmarks](#), [Versioning](#), [Technical Support And Product Updates](#),

Introduction

The TMS320C6000 Network Developer's Kit (NDK) is a platform for development and demonstration of network enabled applications on the TMS320C6000 DSP family. The code included in this NDK release is generic C code which runs on any C64 or C64P device in both big endian and little endian modes. This release does not include any platform specific code; hence the user is required to obtain the appropriate NDK Platform Support Package for his or her desired platform. The NDK includes demonstration software showcasing C6000 DSP capabilities across a range of network enabled applications. In addition, the stack serves as a rapid prototype platform for the development of network and packet processing applications, or to add network connectivity to existing DSP applications for communications, configuration, and control. Using the components provided in the NDK, developers can quickly move from development concepts to working implementations attached to the network. Please check [here](#) for previous release notes and NDK Platform Support Packages. [back to top](#)

Documentation

The following documents provide an overview of the NDK, the networking programming API, and how to port the software to other C6000 based platforms (The following documents are not updated to reflect the changes).

- [The NDK User's Guide](#)
 - Documents the NDK basics, and describes how to develop or port a network enabled DSP application. It also describes how to customize the network environment to fit your DSP environment.
- [The NDK Programmer's Reference Guide](#)
 - Describes the NDK library API calls in detail. It also includes a description of the stack's internal object based API functions.

The NDK Support Package documentation provides detailed information about platform specific details of applicable HAL drivers (Ethernet, serial, timer, and user LED) in a separate document. Please refer to relevant document for details. Release notes from the previous release are available [here](#) [back to top](#)

What's New

1.94.01.001 Release

- Highlights Of Changes In 1.94.01.001
 - Fixes to ensure Big Endian builds work fine.
 - Defects Fixed
 - SDSCM00027183 - NDK 1.94 libraries must be built in big endian
 - SDSCM00027191 - NDK1.94 NSPs for DM642 and DSK6455 should install under NDK_INSTALL_DIR/packages/ti directory
 - SDSCM00027196 - NDK 1.94 NSPs for DM642 and DSK6455 should not have legacy library names changed

[back to top](#)

Upgrading And Compatibility Information

All content in this release is compatible with the previous release. NDK 1.93 drivers and examples packages are compatible with this NDK 1.94.01.001 release.

- This NDK release is only an updated release of the core stack to support Big endian builds. The NDK drivers and examples can be found in the separately released "platform packages."
- The only change in "platform packages" is to add example driver code for NIMU style ethernet driver.
- The NSP (platform packages) now do not contain the LL packet driver libraries anymore. They only support NIMU style drivers. To obtain LL packet drivers for NDK, please use an older NSP release than NDK 1.94.01.001
- The latest "platform packages" for the EVMDM642 and the DSK6455, which contain the drivers and examples, can be downloaded [here](#).
- The "platform packages" for other hardware platforms are released separately.
- Also, please make a special note that from NDK 1.94 onwards, the following IGMP APIs: IGMPJoinHostGroup()/IGMPLeaveHostGroup() have been deprecated. Instead, it is advised to use setsockopt()/getsockopt() APIs with IP_ADD_MEMBERSHIP and IP_DROP_MEMBERSHIP options to achieve the same. Using socket options will enable the application to control multicast communication at socket level.

[back to top](#)

Software Installation Steps

Follow the steps below for NDK software installation:

- Install Code Composer Studio (CCS) using the CCS CD. It must be installed before the NDK software.
- Uninstall any previous version of the NDK, or install in a different directory.
- Install the NDK software using the Stack CD. The installation program will automatically detect where CCS is installed and it will then install the TCP/IP Stack software under the same directory. For example, if CCS is installed in C:\CCStudio_v3.3, the TCP/IP Stack will be installed in C:\CCStudio_v3.3\ndk_1_94_1. The user is given the option of installing in a different directory, if so desired.
- The "NDK_INSTALL_DIR" environment variable must be added to point to top level installation directory, e.g. C:\CCStudio_v3.3\ndk_1_94_1.
- To compile an applicaiton project for Big Endian target, please ensure that the following compilation

flag is defined: **BIGENDIAN**

[back to top](#)

Software Installation Overview

After installation, all the NDK related files are located under `\ndk_1_94_1\packages\ti\ndk` directory, and organized in subdirectories. These subdirectories are fully described in the [NDK User's Guide](#), but can be summarized as follows:

- [\[example \]](#)

The EXAMPLE directory contains sample source code for a variety of applications. Included here are demos that work on Ethernet and/or a serial port. These include initialization examples, HTTP and Telnet servers, configuration examples, and a Serial-PPP to Ethernet router.

 - [\[example\network\]](#) Examples using an Ethernet network. The [\[example\network\<example_name>\common\]](#) directory contains platform independent files which are shared by every platform.
 - [\[example\serial\]](#) Examples using a serial port alone, and with an Ethernet adapter. The [\[example\serial\<example_name>\common \]](#) directory contains platform independent files which are shared by every platform.
 - [\[example\tools\]](#) Utility files common to the example programs. This is a good source for sample code to pull into networking applications. [\[example\tools\common\]](#) directory contains platform independent files which are shared by every platform.
- [\[inc \]](#)

The INC directory contains main include files used in creating network applications.
- [\[lib \]](#)

The LIB directory contains multiple subdirectories for the NDK base library components as well as the HAL libraries.

 - [\[lib\c6400\]](#) This directory contains the c64 version of the NDK libraries and supporting libraries.
 - [\[lib\c6400\all_stk\]](#) This directory contains the c64 version of the NDK stack libraries built with various combinations of features.
 - [\[lib\c6400\hal\]](#) This directory contains the c64 version of the Ethernet, serial, and LED stub drivers, and timer driver libraries.
 - [\[lib\c64plus\]](#) This directory contains the c64plus version of the NDK libraries and supporting libraries.
 - [\[lib\c64plus\all_stk\]](#) This directory contains the c64plus version of the NDK stack libraries built with various combinations of features.
 - [\[lib\c64plus\hal\]](#) This directory contains the c64plus version of the, serial, and LED stub drivers, and timer driver libraries.
- [\[src \]](#)

The SRC directory is the location of the source code to the individual stack components included with the NDK.
- [\[winapps \]](#)

The WINAPPS directory contains small test functions written in C, designed to be executed under Windows using WinSock. These tests communicate with various servers implemented in the DSP example code.
- [\[\doc\stack\]](#)

The DOC directory contains a copy of the NDK documents supplied on the stack installation CD

[back to top](#)

Important First Steps

Of all the [NDK Documentation](#), reading the [NDK User's Guide](#) is the best first step in using the NDK software package. It contains all the necessary instructions for setting up and testing the hardware and software. It also contains an introduction to developing network applications for the DSP environment. The guide's chapters are arranged such that it can be followed in order from *Installation and Test* up through *Network Application Development*. These chapters include:

- Installation and Test of the NDK
- Example Applications
- NDK Library Overview
- Network Application Development

There are also two chapters describing how to tune the NDK for optimal operation in any given DSP environment. These include:

- Network Control Functions
- OS Adaptation Layer

New users should familiarize themselves with the contents of the [NDK User's Guide](#), test their hardware, and try out some of the example applications described within. [back to top](#)

Host Support

- Supported Operating Systems
 - Windows™ XP

[back to top](#)

Dependencies

- Required Software
 - Code Composer Studio Version 3.3 or higher
 - DSP/BIOS 5.31.08 or higher.
 - You may need to download the required GEL file from Spectrum Digital's Web Site, and add these gel files in your CCS configuration to initialize the boards, prior to running any example applications.
 - You may need to download the required BSL library from the Spectrum Digital's Web Site.
- Required Hardware
 - The NDK can run on any platform provided as an "NDK Support Package"
 - In addition to this hardware, a compatible emulator connection to CCS is required. This varies by platform.

[back to top](#)

Device Support

This release supports the following Devices:

| | Runtime Model |
|--------------|-----------------------|
| 64xx | Big And Little Endian |
| 64xx+ | Big And Little Endian |

[back to top](#)

Validation Information

This release was built using the following components:

- C6x Code Generation Tools version 6.0.8
- DSP/BIOS 5.31.08.15

This release was validated using the following components:

- CCS 3.3.49
- C6x Code Generation Tools version 6.0.8
- DSP/BIOS 5.31.08.15

C64x+

- Tested using the EVM6455 board via XDS560 emulator.

C64x

- Tested using the EVMDM642 board via XDS560 emulator.

[back to top](#)

Known Issues

The following issues are known to affect this release:

- SDSCM00025064 - DSP crash on Reboot with a PPPoE server on NDK.

Workaround: Ensure that the PPPoE server of the NDK stack is closed before rebooting the stack.

- SDSCM00024506 - PPPoE client connection handle is not removed after PPPoE session timeout.

Workaround: Ensure that the session timeout on the PPPoE server is configured to be a large value to avoid this issue.

[back to top](#)

Benchmarks

Please refer to the "ReadMe" file for your platform (found in the ReadMe.html file for the platform you are working with, or [here](#)) for specific information on benchmarks.

[back to top](#)

Versioning

This product's version follows a version format, **M.mm.pp.bb**, where **M** is a single digit Major number, **mm** is 2 digit minor number, **pp** is a 2 digit patch number, and **b** is an unrestricted set of digits used as an incrementing build counter. To support multiple side-by-side installations of the product, the product version is encoded in the top level directory, e.g. **ndk_1_94**. Subsequent releases of patch upgrades will be identified by the patch number, e.g. NDK 1.94.01 with directory **ndk_1_94_01**. Typically, these patches only include critical bug fixes. Please note that version numbers and compatibility keys are NOT the same. For an explanation of compatibility keys, please refer to the 'Upgrade and Compatibility Information' section. [back to top](#)

Technical Support And Product Updates

- Please contact softwaresupport@ti.com for technical support.
- Please check [here](#) for previous release notes and NDK Platform Support Packages.

[back to top](#)

Last updated: August 8, 2008