

# Enabling An Existing Program To Use the DSP/BIOS Log Server

**Last Updated: May 10, 2007**

---

This page has been intentionally left blank.

## **IMPORTANT NOTICE**

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI. Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

### Mailing Address:

Texas Instruments  
Post Office Box 655303  
Dallas, Texas 75265

Copyright ©2003, Texas Instruments Incorporated

This page has been intentionally left blank.

## Purpose:

This document describes the steps required to enable an already existing NDK program to use the DSP/BIOS Log Server.

These steps demonstrate to the user how to add the Log Server functionality into the evmdm642 NDK network example “helloWorld.”

## Scope:

These instructions assume that the user’s program already has the NDK stack up and running.

These steps assume that the user has DSP/BIOS Utilities version “1.00”, and all steps are based on the installation directory and tar file name being “biosutils\_1\_00”.

If the user is using a different version of the DSP/BIOS Utilities, then that version string should be substituted into all steps throughout the file. For example, if the user has DSP/BIOS Utilities version 1.00.01, then “biosutils\_1\_00\_01” should be substituted for “biosutils\_1\_00”.

It is also assumed that the user has already installed and properly setup the following required components:

1. Code Composer Studio version 3.2 or 3.3.
2. DSP/BIOS version:
  - a. Recommended: 5.31.04 (or later)
  - b. Supported: 5.31.02 (see work around in step 6.g)
  - c. Not Supported: 5.31.03
3. NDK 1.92
4. Appropriate BSL (Board Support Library) for the hardware being used.
5. WinZip (or equivalent utility) needed to extract tar files.

These steps use the evmdm642 platform as an example. However, these same steps could be applied in the same fashion for any other platform supported by the DSP/BIOS Log Server. Similarly, these steps could be applied to the user’s existing program (that is running the NDK stack) to enable the Log Server in that program.

If the user’s existing program is not currently running the NDK stack, it is recommended that the user refer to the NDK network examples “helloWorld” and “Client” to see how the stack is setup.

## Setup Steps:

1. Copy and extract the Log Server tar.gz file to your hard drive. The following files should be extracted:
  - a. biosutils\_1\_00.tar.gz
2. Once extracted, the following directory will be created: biosutils\_1\_00\packages
3. Create an environment variable to point to the location of the Board Support Library (BSL), which should have been installed already as part of the NDK setup. Add the following environment variable in Windows:

BSL\_EVMDM642\_INSTALLDIR=<bsl install location>/boards/evmdm642  
For example, if you installed the BSL into "C:\CCStudio\_v3.3", then the environment variable would be set as follows:

BSL\_EVMDM642\_INSTALLDIR=C:/CCStudio\_v3.3/boards/evmdm642

Or, for the EVMDM6437:

BSL\_EVMDM6437\_INSTALLDIR=C:/CCStudio\_v3.3/boards/evmdm6437\_v2

4. Create an environment variable to point to the location of BIOS utilities which were just extracted. Add the following environment variable in Windows:

BIOSUTILS\_INSTALL\_DIR=<install\_directory>/biosutils\_1\_00  
For example, if you extracted the biosutils\_1\_00.tar.gz file to "C:\", then the environment variable would be set as follows:

BIOSUTILS\_INSTALL\_DIR=C:/biosutils\_1\_00

5. Next, navigate to the NDK installation examples directory, and make a copy of the NDK network example "helloWorld." This copy will be modified to use the Log Server.
  - a. In Windows Explorer, navigate to the NDK helloWorld example directory: <ndk\_install\_dir>\packages\ti\ndk\example\network\helloWorld
  - b. This directory contains several sub directories that are named according to platform. Make a copy of the directory which corresponds to the platform being used and rename the directory as "logserver." For example, if using the evmdm642 board, copy

<ndk\_install\_dir>\packages\ti\ndk\example\network\helloWorld\evmdm642

and rename this directory to "logserver."

- c. Make a backup copy of the “helloWorld\common” directory. Since some of the files that are common to the helloWorld example must be modified, it is recommended to make a back up copy of the “common” directory.

Copy

```
<ndk_install_dir>\packages\ti\ndk\example\network\helloWorld\common
```

and rename this copy to “common,backup.” This way, the original code of the NDK example will be preserved.

6. Update the example Tconf file to add the required Log Server configuration (Non-RTSC builds only).

- a. This step is only required if the CCS build engine is being used to build the project. If the user is using the RTSC build engine, this step may be skipped.
- b. Edit the file logserver\evmdm642.tcf
- c. Add the following two bold lines of code to this tcf file in order to import the Log Server Configuration:

```
utils.loadPlatform("ti.platforms.evmDM642");  
utils.importFile('helloWorld.tci');  
utils.importFile('Load.tci');  
utils.importFile('LogTrack.tci');
```

7. Update the helloWorld.c file to add the required Log Server C code.

- a. Open the following file for editing:

```
<ndk_install_dir>\packages\ti\ndk\example\network\helloWorld\common\  
helloWorld.c
```

- b. Add the following #include directive to the top of the file:

```
#include <ti/bios/log/ndk/LogServerCgi.h>
```

- c. Update the stack setup function “StackTest()” to add the code required to initialize the LogServerCgi module. The “StackTest()” function contains the NDK stack setup code for the helloWorld example. (Note: If the user is updating their existing program, then it is up to the user to find and update their function which sets up the NDK stack in the same manner as this step describes). Insert the following function call into the beginning of the “StackTest()” function:

```
LogServerCgi_init();
```

- d. Update the stack setup function “StackTest()” to add the code required for creating CGIs. Add the following function call after the DHCP setup code in “StackTest()”:

```
LogServerCgi_add();
```

- e. Update the stack setup function “StackTest()” to add the code required for the HTTP server. This step is only required if the existing program does NOT already have an HTTP server running. Since the helloWorld program does not have an HTTP server, add the following code block after the function call code you just added in the previous step (Please note the curly braces surrounding the code):

```
{
    CI_SERVICE_HTTP    http;

    bzero( &http, sizeof(http) );
    http.cisargs.IPAddr = INADDR_ANY;
    http.cisargs.pCbSrv = &ServiceReport;
    CfgAddEntry( hCfg, CFGTAG_SERVICE,
                CFGITEM_SERVICE_HTTP, 0,
                sizeof(http), (UINT8 *)&http, 0 );
}
```

Note: The above code assumes that the service report callback function is called “ServiceReport.” It is up to the user to modify the code if their service report callback function name is not “ServiceReport.”

- f. Update the stack setup function “StackTest()” to add the code required for removing CGIs. Add the following function call after the while loop which contains the call to NC\_NetStart():

```
LogServerCgi_remove();
```

- g. Optional: Workaround for users of DSP/BIOS 5.31.02: Update the program’s main function() with the following function call in order to work around a known BIOS issue. This step is only required if the user has DSP/BIOS version 5.31.02 or earlier:

```
TRC_disable(TRC_LOGCLK);
```

Note to the user: this step is necessary in order to work around an issue found in DSP/BIOS versions 5.31.02 or earlier. While adding the above code to main() will successfully work around the issue, the recommended solution is for the user to upgrade to DSP/BIOS 5.31.04.



Please also note that DSP/BIOS version 5.31.03 is not supported.

See the DSP/BIOS Utilities release\_notes.html file for more details on the above mentioned issue.

- h. Optional: Add a global struct definition in order to configure some properties of the Log Server. Add the following global definition into helloWorld.c file:

```
LogServerCgi_Config LogServerCgi_config = {
    1, /* CGI thread priority */
    0, /* memory segment for MEM_alloc */
    1024 /* log data buffer size */
};
```

This struct is used to set the values for CGI thread priority, the memory segment used for allocation of the log data buffer (used for transferring log data), and the size of this log data buffer. Note that this step is optional; if this struct is not defined, the Log Server will use default values for these settings.

8. Modify the project to add necessary include paths and libraries. Note that if you are using RTSC to build your application, then you do not need to include the library paths.
  - a. Open the project in Code Composer Studio:  
Open  
`<ndk_install_dir>\packages\ti\ndk\example\network\helloWorld\evmdm642\logserver\helloWorld.pjt`
  - b. Select the menu item “project->build options”
  - c. Under the “compiler” tab, add the following include paths to the project (Not required for RTSC build):

```
-i"%BIOSUTILS_INSTALL_DIR%/packages/ti/bios/log/ndk"
```

- d. Under the “linker” tab, add the following libraries to the project (Not required for RTSC build):

```
-l"%BIOSUTILS_INSTALL_DIR%/packages/ti/bios/log/support/lib/logsupport.a64"
-l"%BIOSUTILS_INSTALL_DIR%/packages/ti/bios/log/ndk/lib/logservercgi.a64"
-l"%BIOSUTILS_INSTALL_DIR%/packages/ti/bios/utils/lib/utils.a64"
```

- e. Under the “DspBiosBuilder” tab, add the following paths to the existing Tconf include path:

```
%BIOSUTILS_INSTALL_DIR%\packages\ti\bios\log\ndk\examples\common;  
%BIOSUTILS_INSTALL_DIR%\packages\ti\bios\log\support;  
%BIOSUTILS_INSTALL_DIR%\packages\ti\bios\utils;
```

9. Build the project
  - a. Select “Project->Rebuild All”
10. Load and run the program
  - a. An IP address will be displayed in standard out.
11. Sanity test the program:
  - a. In a DOS shell, ping the target board:  
type “ping <IP address>”
  - b. The target should respond to the ping command
  - c. In a DOS shell, run the “testudp” NDK program:
    - i. cd to <ndk\_install\_dir>\packages\ti\ndk\winapps
    - ii. type “testudp.exe <IP address>”
    - iii. If the test passes, output similar to the following will be printed:  
“Test loop passed – resetting”
    - iv. Hit “CTRL-C” to terminate the testudp program
12. The program is now ready to work with DVT!
  - a. At this point, DVT can be used to connect to the running target (via the assigned IP address) and used to retrieve log data.