

CodeWarrior™ Development Studio

for Nintendo DSi V1.2 Quick Start for Windows® Operating Systems

SYSTEM REQUIREMENTS

Hardware	PC with 1.4 GHz Intel® Pentium®-compatible processor (or better) 512 MB RAM (1 GB recommended) CD-ROM drive
Operating System	Microsoft® Windows XP SP2 or SP3 Windows Vista® (32 bit) with SP1
Disk Space	2 GB total 500 MB on Windows system disk

This document explains how to install the CodeWarrior for Nintendo DSi V1.2 software on a Windows PC, then create, build, and debug Nintendo DS or DSi projects.

NOTE The software was developed using pre-release product names. Hence, “Nitro” refers to Nintendo DS and “TWL” refers to Nintendo DSi.

NOTE In the procedures that follow advanced users can use the numbered steps. Novices can use the more detailed instructions provided in the substeps.

NOTE The information in this Quick Start may not be latest. See the Developer Notes for the latest information.

Section A: Installing Software

IS Debugger, Nintendo Ensata Emulator, TWL SDK, and CodeWarrior software may be obtained from the Warioworld Software Development Support Group web site: <http://www.warioworld.com>

Install software in the following order:

1. Install most recent IS Debugger software.
2. Install most recent TWL SDK software.
3. Install most recent CodeWarrior Development Studio for Nintendo TWL.
4. Optional for Nintendo DS only. Install most recent Nintendo Ensata Emulator software.

Section B: Install IS Debugger

NOTE While the SDK and IDE can be used to build both DS and DSi projects, install one or both IS Debugger software versions for platforms being targeted.

NOTE CodeWarrior may be installed without installing an IS debugger or TWL SDK first. To install the IS debugger(s) and/or the TWL SDK after CodeWarrior is installed, you must run the batch file {CW}\bin\setTWLSrcTree.bat to create the source trees correctly. See CodeWarrior installation in Section D for details.

1. Install one or both of latest IS Debugger software.
 - a. If you are developing projects for Nintendo DSi, use Nintendo's instructions to install IS-TWL-DEBUGGER software package.
 - b. If you are developing projects for Nintendo DS, use Nintendo's instructions to install IS-NITRO-DEBUGGER software package.

CAUTION Reboot your system after installing debugger software package, even if installer does not prompt you to reboot.

Section C: Install SDK

You must install the SDK for both platforms - Nintendo DS and DSi - before installing CodeWarrior.

1. Following Nintendo's installation instructions, install SDK to an appropriate folder on your hard drive, such as `C:\TwlSDK`

NOTE In previous releases, you had to create the following environment variable that pointed to CodeWarrior tool install folder.

- DS: `CWFOLDER_NITRO`
- DSi: `CWFOLDER_TWL`

Although new CodeWarrior installer now creates this variable for you, you must create it yourself if installing an older release.

1. Open Windows Control Panel
2. Double-click System
3. Select Advanced tab
4. Click Environment Variables
5. Go to the System Variables section, and click New
6. Create the following variable: `TWLSDK_ROOT` with value `C:\TwlSDK`

NOTE For more information about the environment variables, see `QuickStartForSDK.pdf` in the `docs` directory of the TWL SDK.

7. Click OK as many times as necessary to return to Windows

CAUTION You must reboot your system after creating the environment variable.

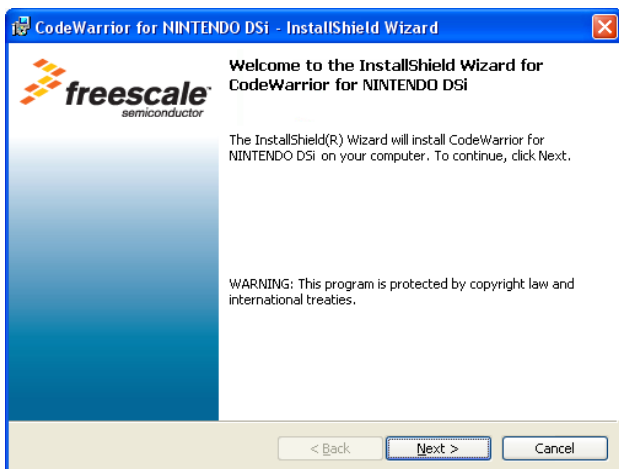
Section D: Install CodeWarrior Development Studio

NOTE CodeWarrior may be installed without first installing IS-NITRO-DEBUGGER or IS-TWL-DEBUGGER. To install the IS debuggers later, follow step 2, below.

1. Install CodeWarrior software

- Obtain **CodeWarrior Development Studio** installer from Nintendo. You must be a Nintendo authorized developer to install this product.
- Start installer — Install wizard starts; welcome window appears

CodeWarrior™ Installer



- Continue to click **Next** to step through installer windows, accepting default settings, until **Select TWL SDK Location** windows appear.
- If installer is unable to locate an installation of an SDK on your system, click **Browse** to locate SDK folder yourself.
- Click **Next** to continue installation process.
- Follow the on-screen instructions to continue the installation process.
- Follow the on-screen instructions to reboot the computer.
- Upon reboot, the installer creates paths to the following source trees:
 - DS: TWLSDK_ROOT and IS_NITRO_DIR
 - DSi: TWLSDK_ROOT and IS_TWL_DEBUGGER_DIR

2. To install an IS debugger after CodeWarrior is installed, you must do the following to complete CodeWarrior installation. This will ensure that your source trees are created correctly.
 - a. Install IS-NITRO-DEBUGGER or IS-TWL-DEBUGGER. This will create the required environment variable:
 - DS: IS_NITRO_DIR
 - DSi: IS_TWL_DEBUGGER_DIR
 - b. Run the following batch file in the CodeWarrior installation directory:
`{CW}\bin\setTWLSrcTree.bat`

Section E: Install Ensata Emulator

The Nintendo Ensata Emulator is a software emulator available for Nintendo DS only.

1. Using Nintendo's installation instructions, install Ensata Emulator to appropriate folder on hard drive, such as:

```
C:\NitroSDK\ensata
```

2. Locate and open CodeWarrior Ensata debugger initialization file `est_cw_debugger.ini` in a text editor. This file is located in `{CodeWarrior}\bin\Plugins\Support\Nitro\IS`
3. Change variable `ensata_path` to that of your `ensata` executable. For example:

```
[control]
ensata_path=C:\NitroSDK\ensata\Release\ensata.exe
```

4. Save `est_cw_debugger.ini` file to disk

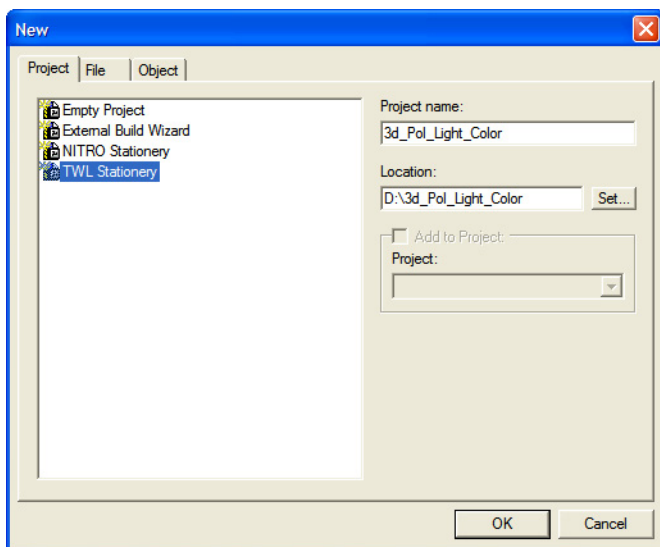
Section F: Creating, Building, and Debugging a Project

1. Create project

To start IDE and display main window, use Windows taskbar to select **Start > Programs > Freescale CodeWarrior > CW for NINTENDO DSI V1.2 > CodeWarrior IDE**

- a. From IDE menu bar, select **File > New** — **New** dialog box appears

New Dialog Box

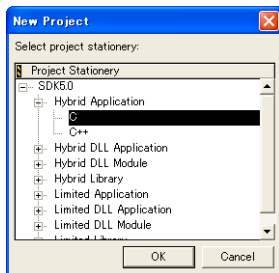


- b. In the left pane, select one of the following, for the platform you are targeting:
- DS: **NITRO Stationery**
 - DSi: **TWL Stationery**
- c. In **Project name** text box, type `3d_Pol_Light_Color`
- d. In **Location** text box, type path at which to create project

NOTE If the **Location** text box does not show the desired project location, click **Set** and then use the **Create New Project** dialog box to select the location you want.

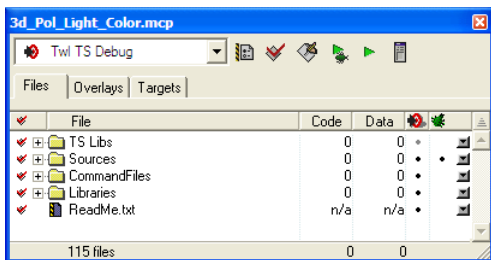
- e. Click **OK** — **New Project** dialog box appears

New Project Dialog Box



- f. Select **SDK5.0 > Hybrid Application > C**
- g. Click **OK** — IDE creates project; `3d_Pol_Light_Color.mcp` project window appears

Project Window



- 2. If you are creating a project for an existing makefile, examine `Makefile` to identify libraries and source files

NOTE Although the project stationery comes preloaded with standard SDK libraries, you must add special libraries yourself.

- a. Open `commondefs.gx.demolib` file in folder `{TwlSDKFolder}\build\buildtools`
- b. Locate line
`LLIBRARIES += libDEMO$(TWL_LIBSUFFIX).a`
This line indicates that you must add a library file (e.g. `libDEMO.TWL.HYB.a`) to your project

- c. Open Makefile file in folder
`{TwlSDKFolder}\build\demos\gx\UnitTours\3D_Pol_LightColor`
- d. Locate line `SRCS = main.c`. This line indicates that you must add `main.c` source file to project.

3. Add source file to project

- a. Locate `main.c` source file in following folder:
`{TwlSDKFolder}\build\demos\gx\UnitTours\3D_Pol_LightColor\src`
- b. Copy this `main.c` file to your `C:\3d_Pol_Light_Color` folder, replacing existing file.

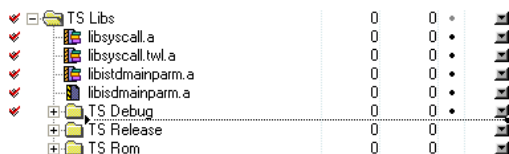
4. Add library files to project

- a. For each {SubDir}, Debug, Release, and Rom; navigate to one of the following files, depending on the platform(s) for which you are building a project.

Platform	File
DS only	<code>{TwlSDKFolder}\build\demos\gx\UnitTours\DEMOLib\lib\ARM9-TS\{SubDir}\libDEMO.a</code>
DSi only	<code>{TwlSDKFolder}\build\demos\gx\UnitTours\DEMOLib\lib\ARM9-TS.LTD\{SubDir}\libDEMO.TWL.LTD.a</code>
DS and DSi	<code>{TwlSDKFolder}\build\demos\gx\UnitTours\DEMOLib\lib\ARM9-TS.HYB\{SubDir}\libDEMO.TWL.HYB.a</code>

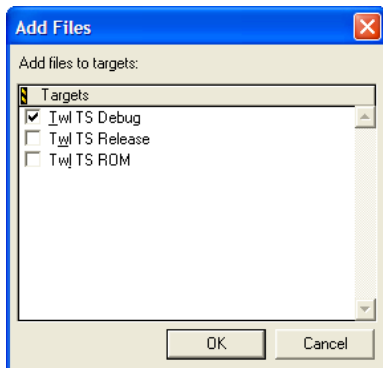
- b. Drag the file to the IDE project window and drop it below **TS Libs > TS {SubDir}** file group — **Add Files** dialog box appears

Drop library under the TS Debug File Group



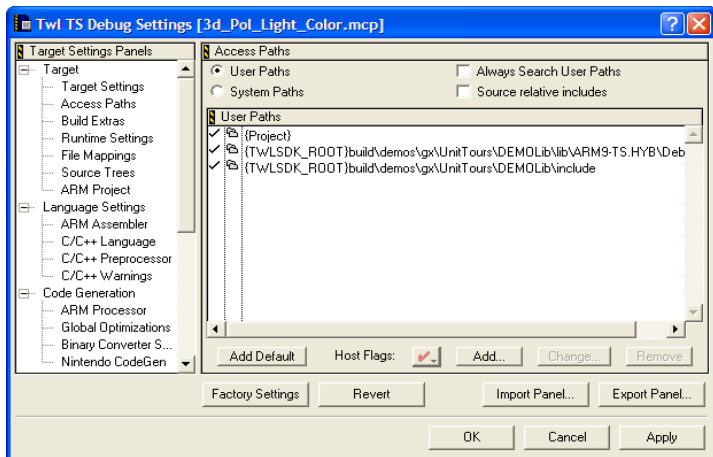
- c. Select **Twl TS {SubDir}** checkbox in **Add Files** dialog box:

Add Files Dialog Box (for DSI)



- d. Click **OK** button:
Respective Library is added to Twl TS {SubDir} target.
- e. Select **Edit > Twl TS {SubDir} Settings** and **Target Settings** window appears.
- f. Select **Target > Access Paths** in Target Settings window

Access Paths Panel



- g. Click **Add - Browse for Folder** dialog box appears
- h. Set **Path Type** to the following: TWLSDK_ROOT
- i. Select the following folder:

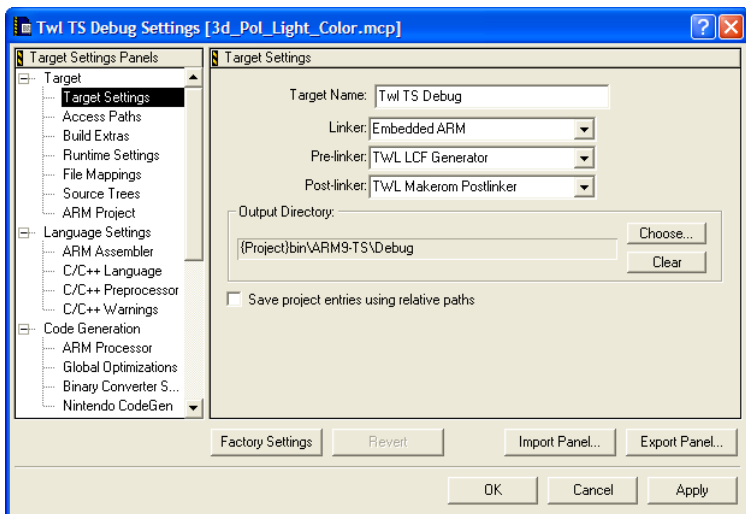
{TWLSDK_ROOT}\build\demos\gx\UnitTours\ DEMOLib\Include

j. Click **OK** - path is added

5. Adjust linker command file settings

- Select **Target > Target Settings** panel
- Set **Pre-linker** to one of the following:
 - DS: **NITRO LCF Generator**
 - DSi: **TWL LCF Generator**

Target Settings Panel (Twi TS Debug)



c. Do one of the following:

- DS: **Select Linker > Nitro LCF Prelinker** panel and verify that **Address** is set at 0x02000000
- DSi: **Select Linker > TWL LCF Prelinker** panel and verify that **Address** is set at 0x02004000

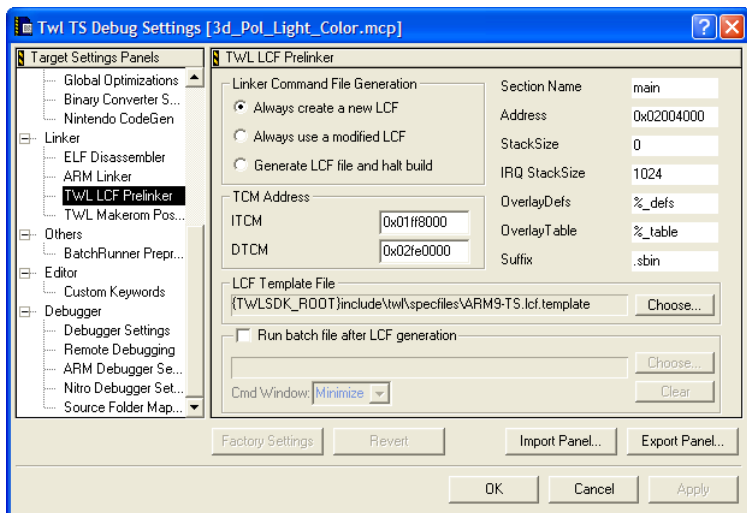
d. Click **OK** button - **Target Settings** window closes

6. Build project

- From IDE menu bar, select one of the following:
 - DS: **Project > Set Default Target > Nitro TS Debug**
 - DSi: **Project > Set Default Target > TWL TS Debug**


- b. From IDE menu bar, select **Project > Make** — IDE updates files and links code into application

TWL LCF Prelinker Target Settings Panel

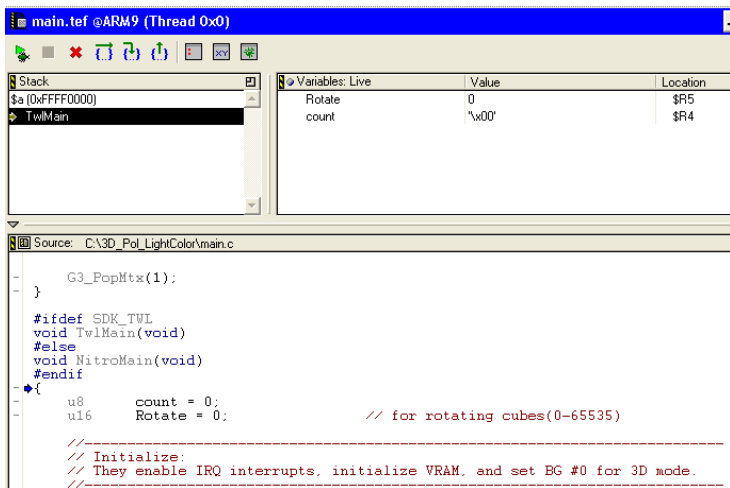



7. Debug project

- a. From IDE menu bar, select **Project > Debug** — IDE assembles, compiles, and links the project; debugger window appears

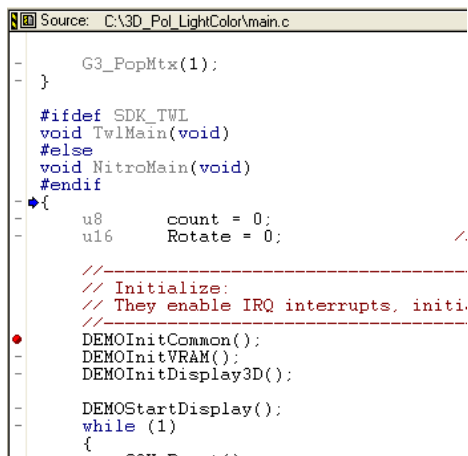
Program counter icon  points to current statement (that is, next statement to be executed)



Debugger Window



- b. In leftmost column of debugger window, click dash next to a statement to set a breakpoint— Breakpoint indicator  appears next to the statement

Setting a Breakpoint



- c. Click Run  button — Processor executes all statements up to (but not including) breakpoint statement and then halts at breakpoint statement
- d. Click Step Over  button a few times to step through the source code
- e. Click Run button again to continue the program execution — LCD displays the program output
- f. From IDE menu bar, select **Debug > Kill** - debug session ends; you may close all open windows

Congratulations!

**You have installed and registered the CodeWarrior software,
and created, built, and debugged a project using the
CodeWarrior IDE.**

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. CodeWarrior is a trademark or registered trademark of Freescale Semiconductor, Inc. in the United States and/or other countries. PROCESSOR EXPERT and EMBEDDED BEANS are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

Copyright © 2007-2009 Freescale Semiconductor, Inc. All rights reserved.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

How to Contact Us

Corporate Headquarters	Freescale Semiconductor, Inc. 6501 William Cannon Drive West Austin, Texas 78735 U.S.A.
World Wide Web	http://www.freescale.com/codewarrior
Technical Support	http://www.freescale.com/support



Revised: 21 September 2009