

Build Procedures for TWL-System Samples Using the IDE

How to Build the TWL-System Samples Using CW IDE

2009/02/09

**The content of this document is highly confidential
and should be handled accordingly.**

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

1	Introduction	6
2	CodeWarrior IDE Configuration	7
3	Building the Archive Sample	8
3.1	Creating a Project for a TWL Application	8
3.2	Preparing the Source and Data of the Samples	9
3.3	Adding the Sample's Source to the Project	9
3.4	Adding TWL-System to the Project.....	10
3.4.1	Configuration of the Access Path for the Debug Version of the Libraries	10
3.4.2	Adding the Debug Versions of the Libraries.....	13
3.5	Adding the Sample's Data to the Project	13
3.6	Preparing to Use Precompiled Headers	14
3.7	When Not Using Precompiled Headers	15
3.8	Building the Project.....	16
4	Building the MultiCellAnimation Sample	17
4.1	Building the g2d Demo Library	17
4.1.1	Creating the Demo Library Project.....	17
4.1.2	Preparing the Source and Data of the Samples.....	18
4.1.3	Adding the Demo Library's Source to the Project	18
4.1.4	Adding TWL-System to the Project	19
4.1.5	Building the Library.....	19
4.2	Building the MultiCellAnimation Sample	19
4.2.1	Creating the MultiCellAnimation Sample Project	19
4.2.2	Adding the Demo Library.....	19
4.2.3	Building the Project	20
4.3	Subprojects	21

Figures

Figure 2-1 IDE Preferences.....	7
Figure 3-1 New Dialog Box	8
Figure 3-2 New Project Dialog Box	9
Figure 3-3 Add Files Dialog Box.....	10
Figure 3-4 Twl TS Debug Settings Dialog Box	11
Figure 3-5 Browse for Folder Dialog Box	12
Figure 3-6 TWL TS Debug Settings Dialog Box	12
Figure 3-7 Twl TS Debug Target Postlinker Settings Panel	14
Figure 3-8 Twl TS Debug Target C/C++ Preprocessor Settings Panel	15
Figure 4-1 Select Project Stationery Selection Screen	17
Figure 4-2 Demo Library Name Change Screen.....	18
Figure 4-3 IDEMultiCellAnimation Project Window	20
Figure 4-4 Configuration of the Targets.....	21

Revision History

Revision Date	Description
2009/02/09	Updated the entire content from NTR to TWL.
2008/03/24	Changed the format of the Revision History.
2007/08/01	Added an explanation of the use of precompiled headers. Added an explanation of <code>nnsys_prefix.h</code> .
2007/02/13	Initial version.

1 Introduction

This document explains the procedure for building the TWL-System samples using the CodeWarrior integrated development environment (IDE).

Chapter 2 CodeWarrior IDE Configuration describes the required common settings when building the TWL-System samples with CodeWarrior.

Chapter 3 Building the Archive Sample describes settings and building procedures for simple projects using the TWL-System library.

Chapter 4 Building the MultiCellAnimation Sample describes settings and building procedures for projects using private libraries.

For specific details about the CodeWarrior IDE, see the CodeWarrior user manuals.

2 CodeWarrior IDE Configuration

The CodeWarrior IDE settings must first be configured.

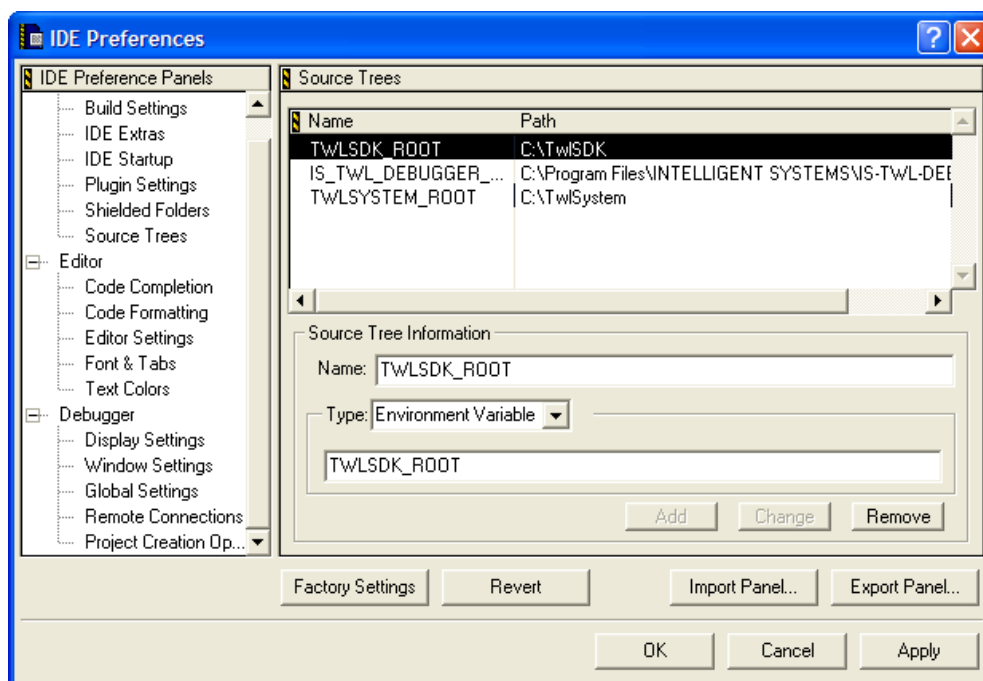
To build and run the TWL-System samples, you must have TWL-SDK and either the IS-TWL-DEBUGGER software and hardware or IS-NITRO-EMULATOR and IS-NITRO-DEBUGGER, as well as TWL-System. In CodeWarrior IDE, you must configure the location where these programs are installed as the source tree. If the command-line build environment has already been set up, the location where these programs are installed has already been configured in the environment variables. You can configure the source tree in CodeWarrior IDE using the values of these environment variables. This guide assumes the use of IS-TWL-DEBUGGER.

To configure the source tree for TWL-SDK, do the following.

1. Run the IDE for CodeWarrior for Nintendo DSi Version 1.1.
2. On the **Edit** menu, select **Preferences** to open the **IDE Preferences** dialog box.
3. In the **IDE Preference Panels**, select **General – Source Trees**.
4. Under **Source Tree Information**, in the **Name** box, type **TWLSDK_ROOT**.
5. In the **Type** box, select **Environment Variable**, type the environment variable name **TWLSDK_ROOT** below that, and then click **Add**.

You should also register the source trees for `IS_TWL_DEBUGGER_DIR` and `TWLSYSTEM_ROOT` in the same way. To apply the modified settings, click **OK** or **Apply** in the **IDE Preferences** dialog box.

Figure 2-1 IDE Preferences



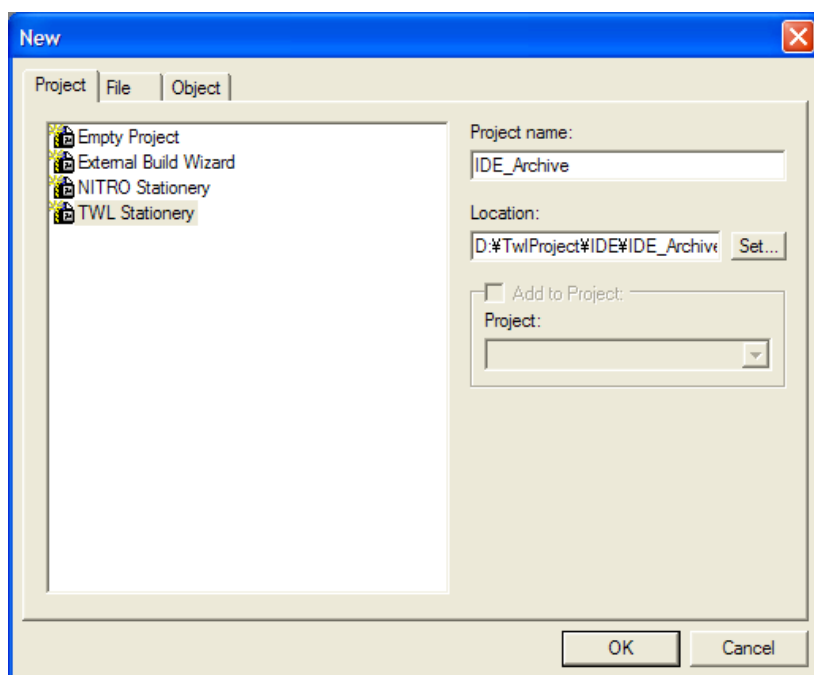
3 Building the Archive Sample

This section explains the procedure for building the Archive sample of TWL-System's fnd library using CodeWarrior IDE. The procedures are for building the Archive sample as a HYBRID version application. Note that for NITRO or LIMITED version applications, the selected stationary and library are different.

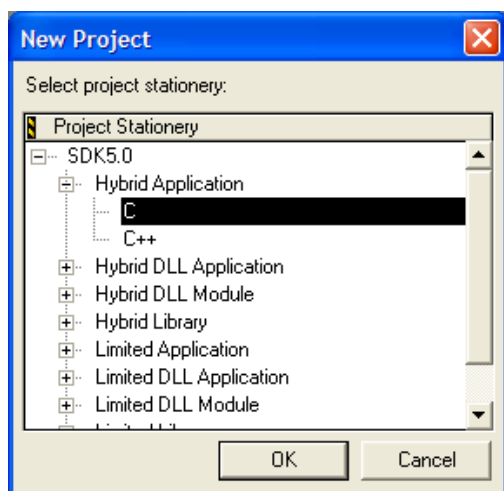
3.1 Creating a Project for a TWL Application

1. In the IDE, on the **File** menu, select **Create New** to display the **New** dialog box.

Figure 3-1 New Dialog Box



2. On the **Project** tab, select **TWL Stationery**, enter the **Project name** and **Location** of the project, and then click **OK**. The **New Project** dialog box opens.
3. In the **New Project** dialog box, under **Project Stationery**, select **SDK5.0 – Hybrid Application – C** and then click **OK** to create the project.

Figure 3-2 New Project Dialog Box

3.2 Preparing the Source and Data of the Samples

Having created a project for the Archive sample, you can now prepare the source and data of the `fnd` library's Archive sample for the project.

First, the `main.c` file that has been added to the project in advance by the stationery is not used, so delete it from the project directory using Windows Explorer. Next, copy the `include`, `src`, and `data` directories from the `TwlSystem/build/demos/fnd/archive` directory of TWL-System to the project directory.

At this point the preparations required for building and executing the samples have been completed. Next, these files are registered in the project.

3.3 Adding the Sample's Source to the Project

In the Sources group of the project, add the source of the sample that was copied into the project directory.

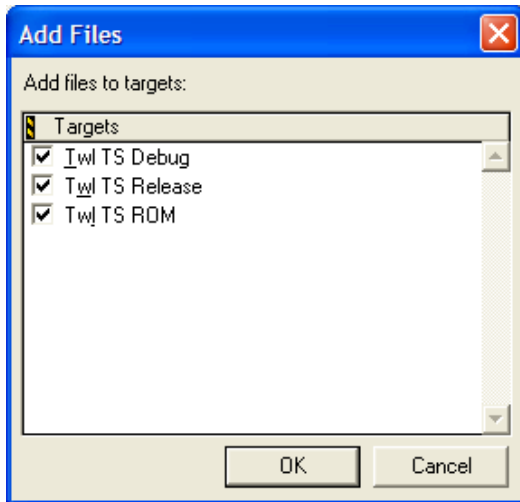
Adding files to a project requires two operations: configuring the access path to the files being added and then adding the files. However, under the project directory (the directory where the MCP file is located), the access path is configured when a project is created. So there is no need to configure the access path again.

First, because it is not required, you can remove the `main.c` file that was added in advance by the stationery. Then you can add the files.

1. In the CodeWarrior IDE project window, in the Sources group, right-click `main.c` and then click **Remove**.
2. In the project window, right-click the Sources group and then click **Add Files** to open the **Select files to add** dialog box.

3. From the `src` directory that was just copied, select `main.c`, `nns_util.c`, and `sdk_init.c` and then click **Open** to display the **Add Files** dialog box. The sample's source is added to the targets **Twl TS Debug**, **Twl TS Release**, and **Twl TS ROM**.
4. In the **Add Files** dialog box, verify that all three targets are selected and then click **OK**.

Figure 3-3 Add Files Dialog Box

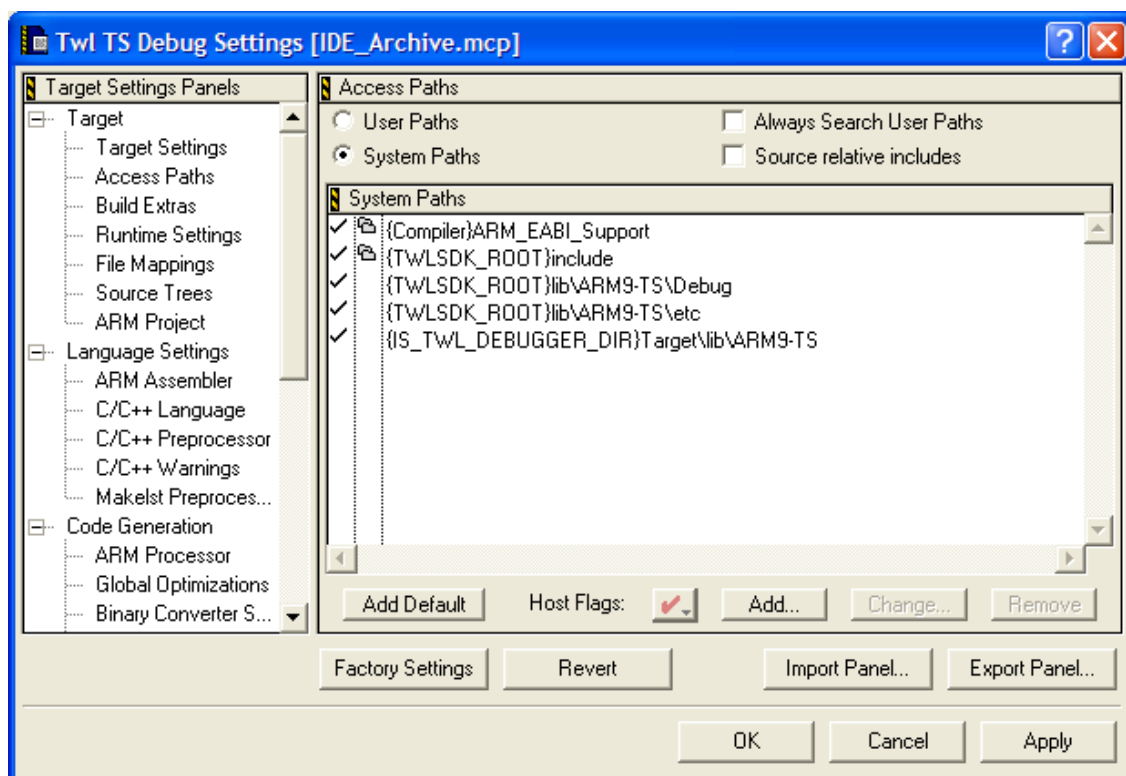


3.4 Adding TWL-System to the Project

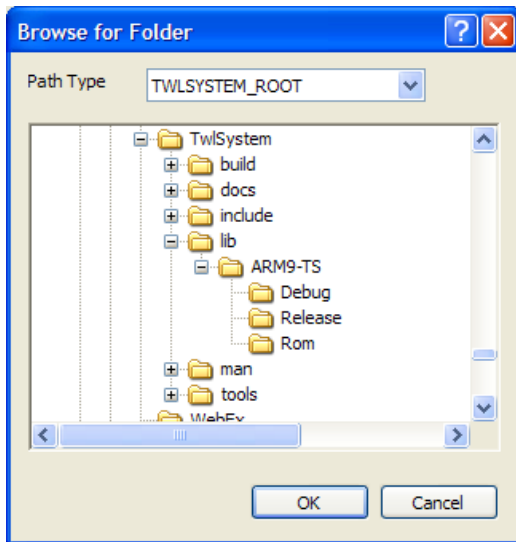
You can now add TWL-System to the project. This section explains how to configure the **Twl TS Debug** target. The configuration method is the same used for the other targets, so they should be configured in the same way.

3.4.1 Configuration of the Access Path for the Debug Version of the Libraries

1. Verify that the project's target is **Twl TS Debug**, and then on the **Edit** menu, select **Twl TS Debug Settings** to open the **Twl TS Debug Settings** dialog box.

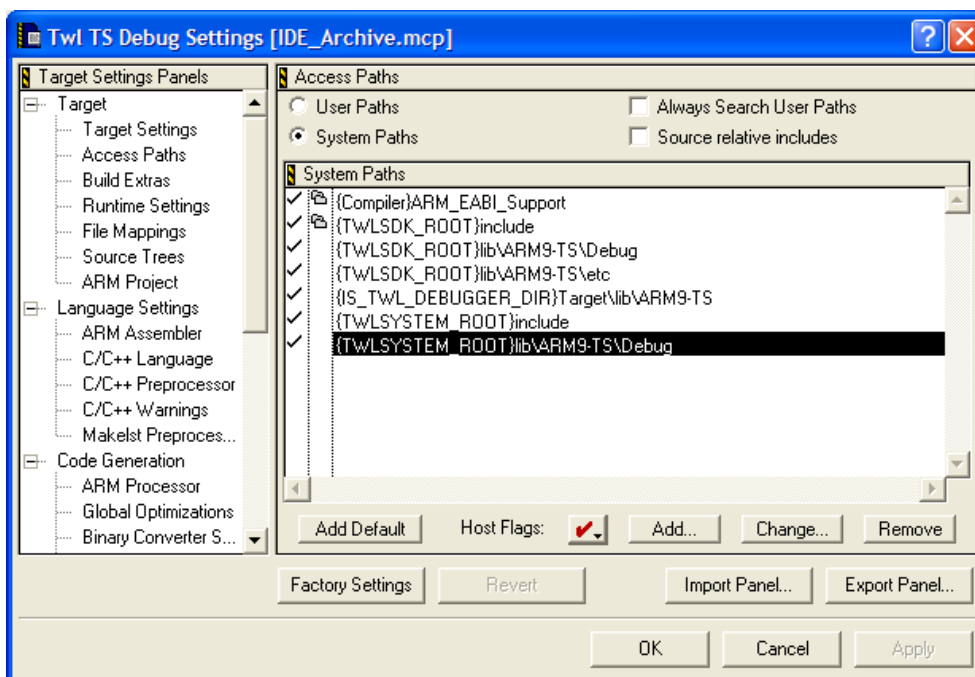
Figure 3-4 Twl TS Debug Settings Dialog Box

2. In the **Target Settings Panels**, select **Target – Access Paths** to display the access path panel on the right.
3. To add the TWL-System access path to the system path, under **Access Paths**, click **System Paths** and then click **Add** to open the **Browse for Folder** dialog box.
4. In the **Path Type** list, click **TWLSYSTEM_ROOT**, select the directory that contains the TWL-System header files (TwlSystem/include) and then click **OK** to add "{TWLSYSTEM_ROOT}include" to the system path. Click **OK** or **Apply** to apply the setting changes to the project.

Figure 3-5 Browse for Folder Dialog Box

A folder icon is displayed at the beginning of the access path that was added. This icon indicates that the path is going through all directories under the access path that has been configured. In the command-line build environment, the path only goes through the `include` directory. If you want to make the environment match it strictly, click the icon and delete it.

Add the `TwlSystem/lib/ARM9-TS/Debug` directory, which stores the debug version of the library, to the system path in the same way.

Figure 3-6 TWL TS Debug Settings Dialog Box

3.4.2 Adding the Debug Versions of the Libraries

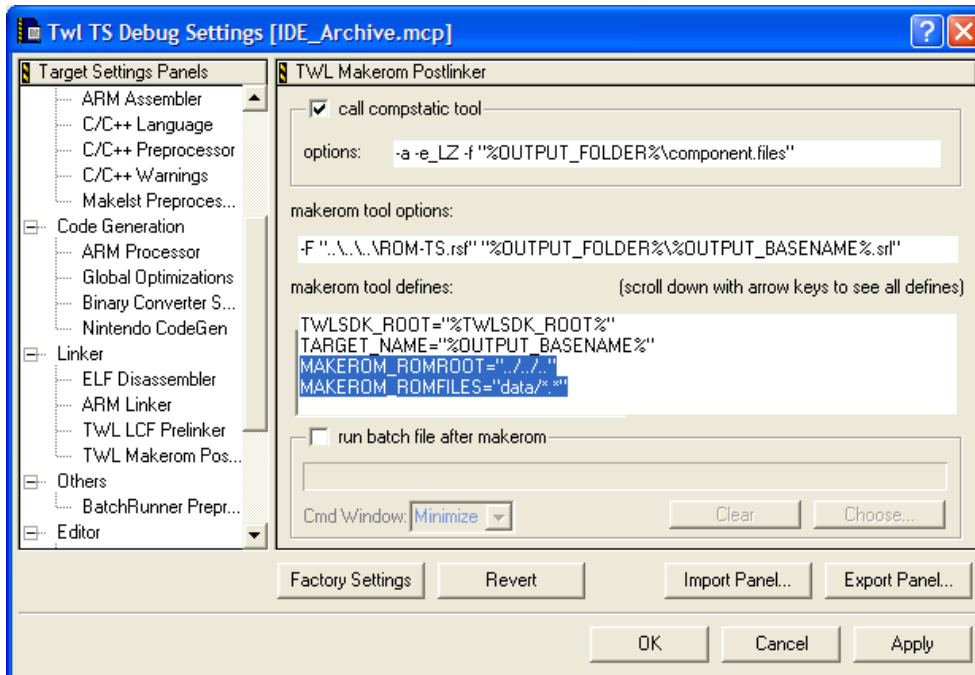
Create a group with the same structure as the TS Libs group that has been added to the TWL-SDK libraries and then add the TWL-System libraries inside this group.

1. On the **Project** menu, select **Create Group** to display the group creation dialog box.
2. Enter **TwlSystem** and then click **OK** to create a group named **TwlSystem** in the project file list.
3. Create a **TS Debug** group inside the **TwlSystem** group in the same way.
4. Right-click the **TwlSystem/TS Debug** group and then click **Add Files** to add the following six **TwlSystem/lib/ARM9-TS/Debug** libraries.
 - `libnnssnd.TWL.HYB.a`
 - `libnnsfnd.TWL.HYB.a`
 - `libnnsg2d.TWL.HYB.a`
 - `libnnsg3d.TWL.HYB.a`
 - `libnnsgfd.TWL.HYB.a`
 - `libnnsmcs.TWL.HYB.a`
5. In the **Add Files** dialog box, select only the **Twl TS Debug** target and then click **OK**.

3.5 Adding the Sample's Data to the Project

Register the data used by the sample program in the project.

1. Verify that the project target is **Twl TS Debug**, and then on the **Edit** menu, click **Twl TS Debug Settings**.
2. In the **Target Settings Panels** list, click **Linker – TWL Makerom Postlinker** to display the **TWL Makerom Postlinker** panel.

Figure 3-7 Twl TS Debug Target Postlinker Settings Panel

3. In the **makerom tool defines** box, change the settings of **MAKEROM_ROMROOT** and **MAKEROM_ROMFILES** as follows.

```
MAKEROM_ROMROOT= "..../.."
MAKEROM_ROMFILES= "data/*. *"
```

4. In **MAKEROM_ROMROOT**, configure the directory where there is data using the relative path from the directory in which SRL files are output. Also, in **MAKEROM_ROMFILES**, specify the files to include in the ROM.
5. Click **OK** or **Apply** to apply the setting changes to the project.

3.6 Preparing to Use Precompiled Headers

By default, precompiled headers are configured for use when you build a project using the Hybrid Application project stationery. Two precompiled headers are registered to the project: one in C, the other in C++.

Add the TWL-System header file to the precompiled headers, which are `Twl_TS.pch` (in C) and `Twl_TS.pch++` (in C++).

```
/* include header for TWL-System */
#include <nnsys_prefix.h>
#include <nnsys.h>
```

The `nnsys_prefix.h` file defines the macros that are passed to the compiler in the `make` build environment. Including this header file allows you to compile with the same settings as builds using `make`.

Additionally, when using precompiled headers, you can delete the references to TWL-System header files within the source code and the demo code header files, because there is no need to include them.

3.7 When Not Using Precompiled Headers

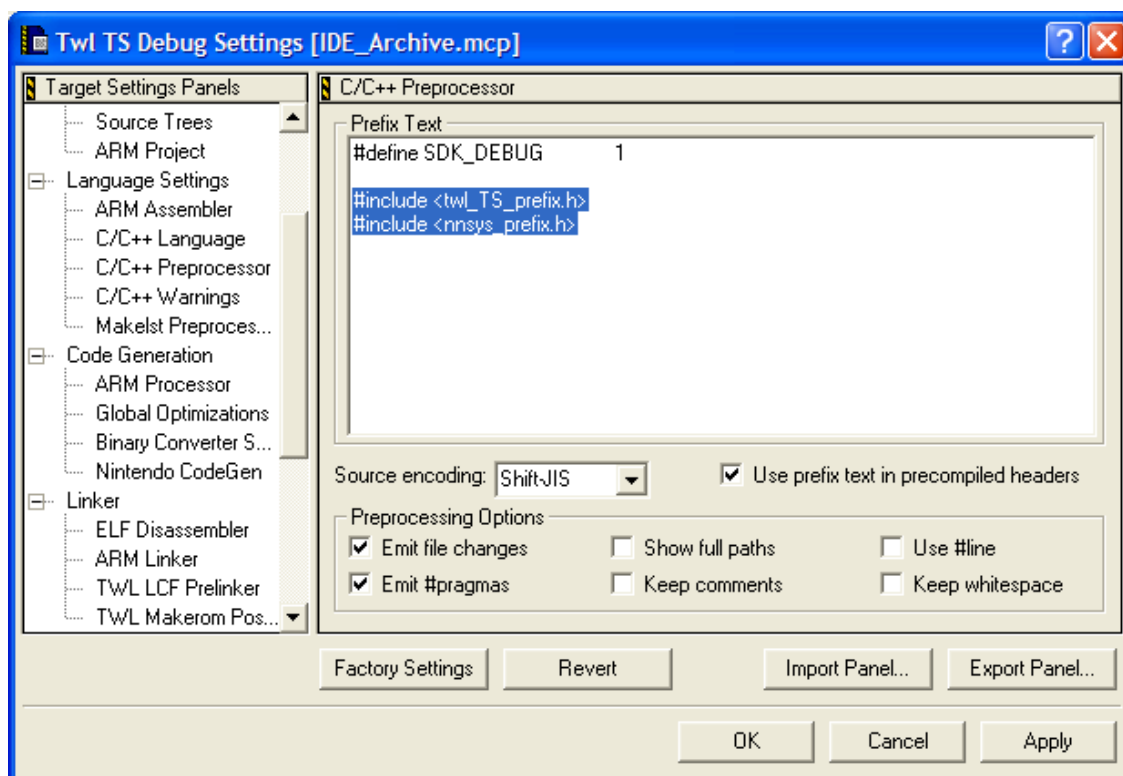
1. To avoid the use of precompiled headers, remove the following three files from your project.
 - Twl_TS.pch
 - Twl_TS.h
 - Twl_TS.pch++
2. In the **Twl TS Debug Settings** dialog box, make the following changes in the **Prefix Text** box for the **Language Settings – C/C++ Preprocessor**.
 - Delete the line that reads:

```
#include "Twl_TS.h"
```
 - Remove the comment-out marks from around the line that reads:

```
#include <twl_TS_prefix.h>
```
 - Add the following below that:

```
#include <nnsys_prefix.h>
```

Figure 3-8 Twl TS Debug Target C/C++ Preprocessor Settings Panel



3.8 Building the Project

At this point the preparations for building the Archive sample have been completed.

Verify that the project target is **Twl TS Debug**. Then, on the **Project** menu, click **Make** (or press the F7 key) and build the target.

The method for configuring the **Twl TS Release** target and the **Twl TS ROM** target is the same as for **Twl TS Debug**. Configure the projects in the same way and build them.

4 Building the MultiCellAnimation Sample

This section explains the procedure up through the building of the MultiCellAnimation sample of the TWL-System g2d library. This is an example of using a private library.

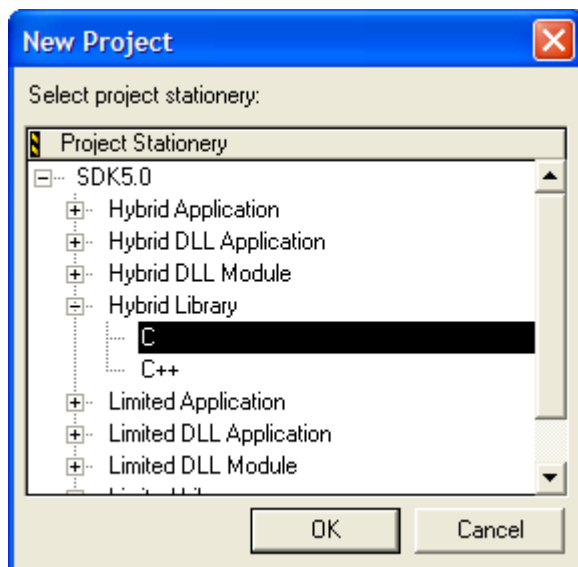
4.1 Building the g2d Demo Library

The g2d library samples use a private demo library unique to g2d samples called `libg2d_demo.a`. First, create a project for this demo library.

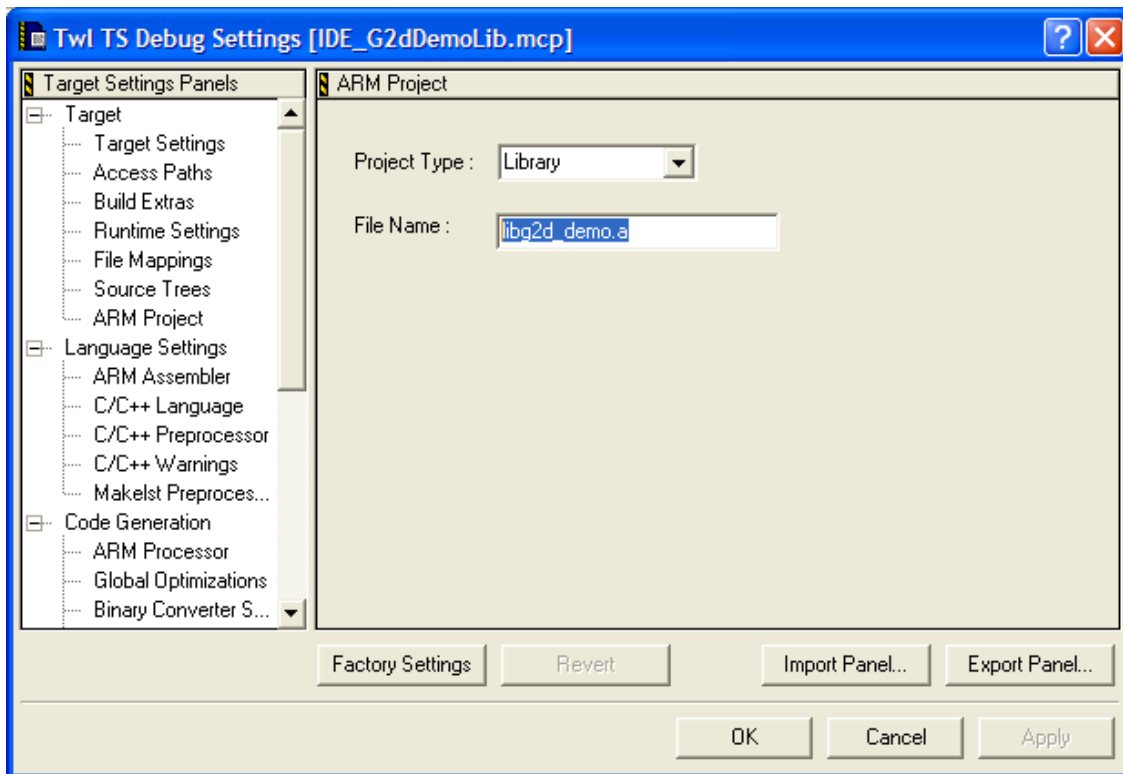
4.1.1 Creating the Demo Library Project

1. In CodeWarrior IDE, on the **File** menu, click **Create New** to open the **New** dialog box.
2. On the **Project** tab, select **TWL Stationery**, enter the **Project name** and **Location** of the project, and then click **OK**. The **New Project** dialog box opens.
3. In the **New Project** dialog box, under **Project Stationery**, select **SDK5.0 – Hybrid Library – C** and then click **OK** to create the project.

Figure 4-1 Select Project Stationery Selection Screen



Verify that the project target is **Twl TS Debug**, and when **Twl TS Debug Settings** is selected from the **Edit Menu**, the **Twl TS Debug Settings** dialog box appears. When **Target-ARM Project** is selected in the **Target Settings** panel in this dialog box, the **ARM Project** panel appears on the right. To change the name of the demo library from the default, change **File Name** to `libg2d_demo.a`. Click **OK** or **Apply** to apply the setting changes to the project.

Figure 4-2 Demo Library Name Change Screen

4.1.2 Preparing the Source and Data of the Samples

Having created a project for the demo library, you can now prepare the source of the g2d samples' demo library.

First, the `main.c` file that has been added to the project in advance by the stationery is not used, so use Windows Explorer to delete it from the project directory. Next, copy the two directories `include` and `src` from the `TwlSystem/build/demos/g2d/demolib` directory of TWL-System to the project directory.

4.1.3 Adding the Demo Library's Source to the Project

Add the demo library's source to the **Sources** group in the project. The method for adding the source is the same as for the Archive sample.

First, delete the `main.c` file added by the stationery from the project, because it is not required. Next, add the four files `fontData.c`, `loader.c`, `print.c`, and `system.c` in the `src` directory that was just copied to the project directory. In the **Add Files** dialog box, be sure to select all three targets.

4.1.4 Adding TWL-System to the Project

You can now add TWL-System to the project. This section explains how to configure the **Twl TS Debug** target. The configuration method is the same for the other targets, so they should be configured in the same way.

There is no need to add the TWL-System library binaries (.a files) when building the demo library. Only configure the access path to the TWL-System `include` directory. The configuration method is the same as for the Archive sample.

In the **Twl TS Debug Settings** dialog box, in the **Target Settings Panels** list, select **Targets – Access Paths** and then add the directory that contains the TWL-System header files (`TwlSystem/include`) to the system path.

4.1.5 Building the Library

At this point the preparations for building the demo library have been completed. Verify that the project target is **Twl TS Debug**. Then, on the **Project** menu, select **Make** (or press F7) and build the target.

Build the **Twl TS Release** target and the **Twl TS ROM** target by performing the same configuration.

4.2 Building the MultiCellAnimation Sample

This section explains how to build the g2d library sample, MultiCellAnimation.

4.2.1 Creating the MultiCellAnimation Sample Project

The method for creating the MultiCellAnimation sample project is the same as the method for creating the Archive sample project. Create the project following the creation method for the Archive sample project, and then add the sample source, data, and TWL-System libraries.

4.2.2 Adding the Demo Library

The g2d library samples use the g2d demo library, so the demo library must also be added to the project. This section explains the configuration method for the **Twl TS Debug** target. The configuration method is the same for the other targets, so configure them the same way.

4.2.2.1 Configuration of the Access Path for the Demo Library

You can now register the demo library in the user access paths. In the **Twl TS Debug Settings** dialog box, in the **Target Settings Panels** list, select **Targets – Access Paths** and then add the `include` directory of the g2d demo library and the directory where the debug version of the library is stored (`bin/ARM9-TS.HYB/Debug` in the demo library project) to the user paths.

4.2.2.2 Registering the Debug Version of the Demo Library

Create a group with the same structure as the **TS Libs** group that has the TWL-SDK libraries added to it, and add the g2d demo library inside this group.

1. On the **Project** menu, click **Create Group** to open the group creation dialog box.

2. Enter **G2dDemoLib** and then click **OK** to create a group named **G2dDemoLib** in the project file list. Create a **TS Debug** group inside the **G2dDemoLib** group in the same way.
3. Right-click the **G2dDemoLib/TS Debug** group and then click **Add Files** to add the file `bin/ARM9-TS.HYB/Debug/libg2d_demo.a` in the project directory of the demo library.
4. In the **Add Files** dialog box, select only by the **Twl TS Debug** target and then click **OK**.

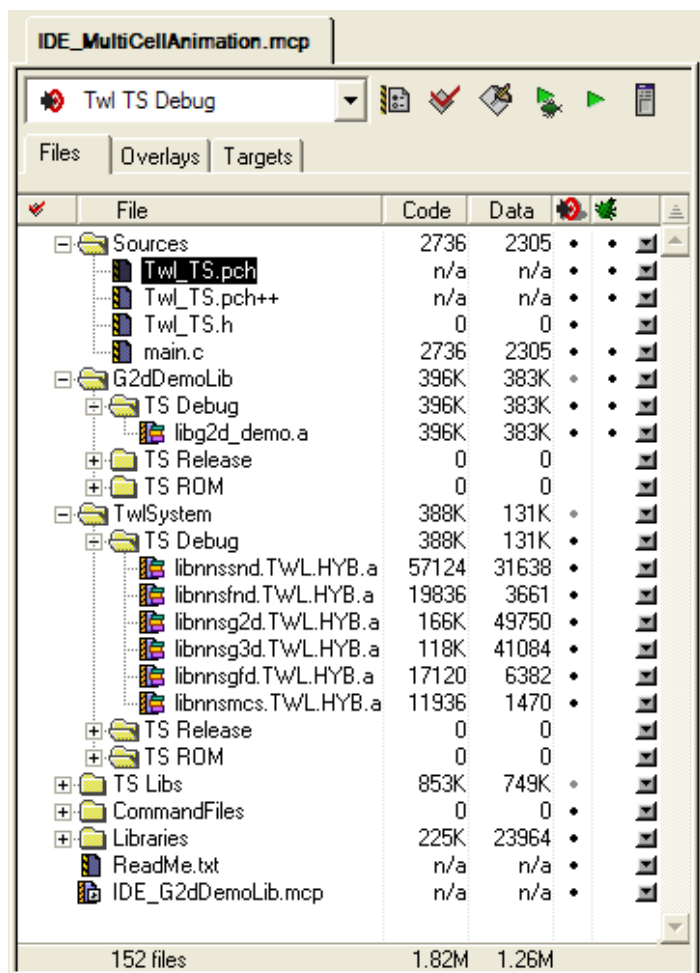
4.2.3 Building the Project

At this point, the preparations for building the MultiCellAnimation sample have been completed. Verify that the project target is **Twl TS Debug**. Then, on the **Project** menu, select **Make** (or press F7) and build the target.

Build the **Twl TS Release** target and the **Twl TS ROM** target by performing the same configuration.

Figure 4-3 shows the contents of the IDE_MultiCellAnimation project window.

Figure 4-3 IDEMultiCellAnimation Project Window



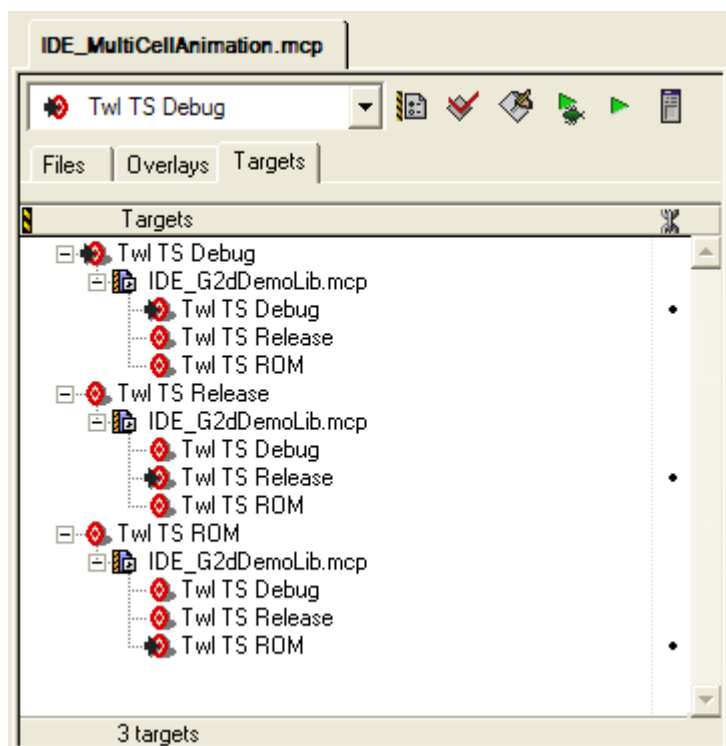
4.3 Subprojects

In Figure 4-3, the g2d demo library project is being added to the MultiCellAnimation project as a subproject. Doing so this way will trigger the demo library to be built first when building the MultiCellAnimation project if the g2d demo library must be built.

Adding the g2d demo library project is possible with the same method as adding files to a project. To link and build, it will be necessary to configure the target after adding it. The target settings are configured in the project's target tab.

When building each target of the sample, configure which targets of the demo library to build in the target tab. Normally, when building the **Twl TS Debug** target of the sample, it is okay to build the **Twl TS Debug** target of the demo library, so click the target icon before the **Twl TS Debug** target of the demo library to add an arrow.

Figure 4-4 Configuration of the Targets



CodeWarrior is a registered trademark of FreeScale Semiconductor, Inc.

All other company and product names in this document are the trademarks or registered trademarks of their respective companies.

© 2007-2009 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.