

# IDE でのサンプルのビルド手順

## NITRO-System サンプルを CW IDE でビルドする方法

Ver 1.1.0

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、  
厳重な取り扱い、管理を行ってください。

## 目次

1	はじめに .....	5
2	CodeWarrior IDEの環境設定 .....	5
3	Archiveサンプルのビルド .....	6
3.1	NITRO用アプリケーションのプロジェクトの作成 .....	6
3.2	サンプルのソースとデータの準備 .....	7
3.3	サンプルのソースをプロジェクトに追加 .....	7
3.4	NITRO-Systemをプロジェクトに追加 .....	8
3.4.1	デバッグ版のライブラリのアクセスパス設定 .....	8
3.4.2	デバッグ版のライブラリの追加 .....	10
3.5	サンプルのデータをプロジェクトに追加 .....	10
3.6	プリコンパイルヘッダ使用の準備 .....	11
3.7	プリコンパイルヘッダを使用しない場合 .....	12
3.8	プロジェクトのビルド .....	12
4	MultiCellAnimationサンプルのビルド .....	13
4.1	g2dデモライブラリのビルド .....	13
4.1.1	デモライブラリプロジェクトの作成 .....	13
4.1.2	サンプルのソースとデータの準備 .....	13
4.1.3	デモライブラリのソースをプロジェクトに追加 .....	14
4.1.4	NITRO-Systemをプロジェクトに追加 .....	14
4.1.5	ライブラリのビルド .....	14
4.2	MultiCellAnimationサンプルのビルド .....	14
4.2.1	MultiCellAnimationサンプルのプロジェクト作成 .....	14
4.2.2	デモライブラリの追加 .....	14
4.2.2.1	デモライブラリのアクセスパスの設定 .....	14
4.2.2.2	デバッグ版デモライブラリの登録 .....	15
4.2.3	プロジェクトのビルド .....	15
4.3	サブプロジェクト .....	16



図 2-1	IDEの環境設定 .....	5
図 3-1	新規ダイアログ .....	6
図 3-2	アプリケーションステーションナリの選択画面 .....	6
図 3-3	ソースファイルの追加画面 .....	7
図 3-4	Nitro TS Debugの設定ダイアログ .....	8
図 3-5	インクルードディレクトリの選択画面 .....	9
図 3-6	Nitro TS Debugターゲットのアクセスパス設定画面 .....	9
図 3-7	Nitro TS DebugターゲットのPostlinker設定画面 .....	11
図 3-8	Nitro TS DebugターゲットのC/C++ Preprocessor設定画面 .....	12
図 4-1	ライブラリステーションナリの選択画面 .....	13

図 4-2	MultiCellAnimationプロジェクトの内容 .....	15
図 4-3	ターゲットの設定 .....	16

## 改訂履歴

版	改訂日	改 訂 内 容	承認者	担当者
1.1.0	2007-08-01	<ul style="list-style-type: none"> <li>•プリコンパイルヘッダの使用に関する説明を追記。</li> <li>•nnsys_prefix.h に関する説明を追記。</li> </ul>		西田泰
1.0.0	2007-02-13	初版。		西田泰

# 1 はじめに

このドキュメントでは、NITRO-System のサンプルを CodeWarrior IDE を使用してビルドを行う場合の手順を説明しています。CodeWarrior IDE の詳しい情報は、CodeWarrior のユーザマニュアルをご覧ください。

## 2 CodeWarrior IDE の環境設定

まず最初に CodeWarrior IDE の環境の設定を行います。

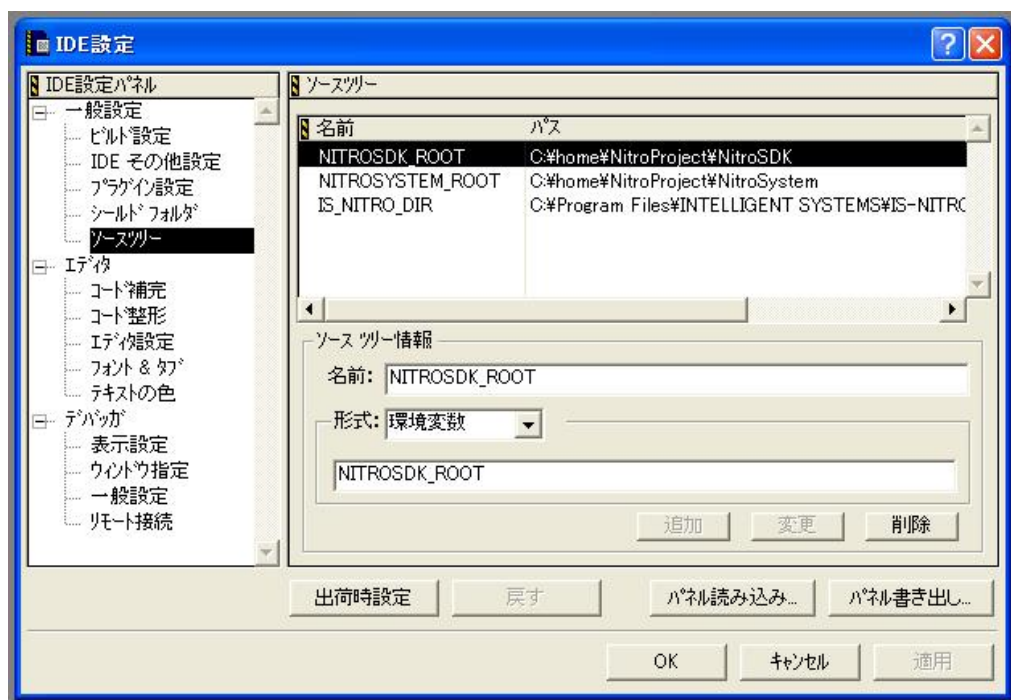
NITRO-System のサンプルのビルドと実行には NITRO-SDK、IS-NITRO-DEBUGGER、NITRO-System が必要です。これらがインストールされている場所をソースツリーとして CodeWarrior IDE に設定します。既にコマンドラインでのビルド環境が整っている場合、これらがインストールされている場所は環境変数に設定されていると思います。この環境変数の値を用いて CodeWarrior IDE にソースツリーを設定します。

まず NITRO-SDK のソースツリーを設定します。設定の手順は下記の通りです。

- CodeWarrior for NINTENDO DS Version2.0 の IDE を起動します。
- 編集メニューから[環境設定]を選択しますと、IDE 設定ダイアログが表示されます。
- IDE 設定パネルから[一般設定]－[ソースツリー]の項目を選択します。
- ソースツリー情報の名前入力欄に「NITROSDK\_ROOT」を入力し、形式として[環境変数]を選択し、その下に環境変数名「NITROSDK\_ROOT」を入力して[追加]ボタンを押します。

同様に、「NITROSYSTEM\_ROOT」と「IS\_NITRO\_DIR」のソースツリーも登録してください。

図 2-1 IDE の環境設定



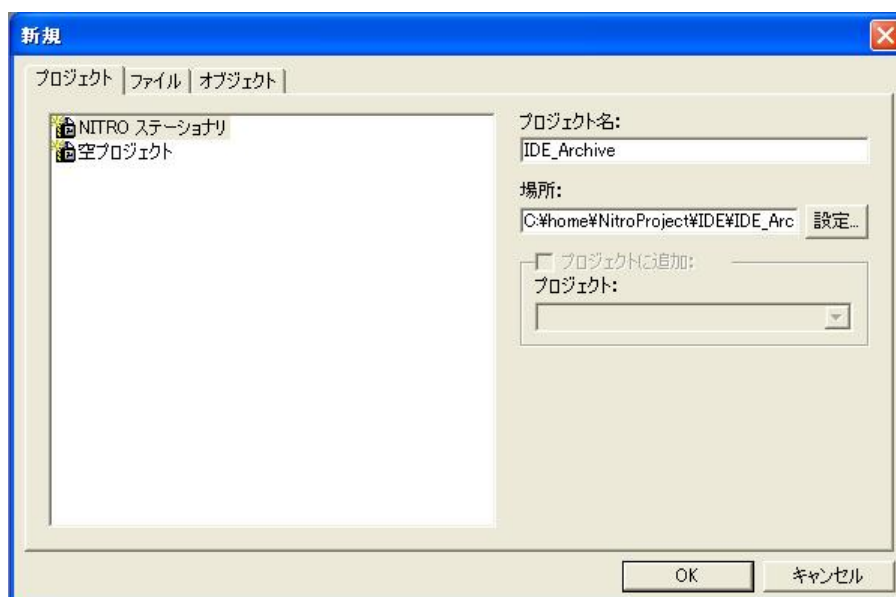
## 3 Archive サンプルのビルド

この章では、CodeWarrior IDE を使って、NITRO-System の fnd ライブラリの Archive サンプルをビルドするまでの手順を説明します。

### 3.1 NITRO 用アプリケーションのプロジェクトの作成

IDE のファイルメニューから[新規]を選択しますと、新規ダイアログが表示されます。この新規ダイアログのプロジェクトタブの中の[NITRO ステーションナリ]を選択し、右の欄にプロジェクト名と場所を入力して[OK]ボタンを押してください。

図 3-1 新規ダイアログ



[OK]ボタンを押しますと、新規プロジェクトダイアログが表示されます。このダイアログのプロジェクトステーションナリで[SDK3.0]－[ARM9-TS Application]－[C]を選択し、[OK]ボタンを押しますと、プロジェクトが作成されます。

図 3-2 アプリケーションステーションナリの選択画面



## 3.2 サンプルのソースとデータの準備

Archive サンプル用のプロジェクトができましたので、fnd ライブラリの Archive サンプルのソースとデータをプロジェクト用に準備します。

まず最初に、ステーションナリによって予めプロジェクトに追加されている `main.c` は使用しませんのでプロジェクトのディレクトリから削除してください。次に NITRO-System の `NitroSystem/build/demos/fnd/archive` ディレクトリから、“include”、“src”、“data”の3つのディレクトリをプロジェクトのディレクトリにコピーします。

これでサンプルのビルドと実行に必要なファイルの準備ができました。次にこれらのファイルをプロジェクトに登録します。

## 3.3 サンプルのソースをプロジェクトに追加

プロジェクトのディレクトリにコピーしたサンプルのソースをプロジェクト内の Sources グループ内に追加します。

プロジェクトへのファイルの追加は、追加するファイルへのアクセスパスの設定と、ファイルの追加の2つの作業が必要ですが、プロジェクトのディレクトリ(.mcp ファイルがあるディレクトリ)以下は、プロジェクト作成時にアクセスパスが設定されていますので、新たにアクセスパスを設定する必要はありません。

まず、ステーションナリにより予め追加されている `main.c` は必要ありませんので削除します。CodeWarrior IDE のプロジェクトウィンドウにあります Sources グループ内の `main.c` の上でマウスの右ボタンをクリックし、表示されるメニューから[削除]選びますと、そのファイルが削除されます。

次にプロジェクトウィンドウにあります Sources グループの上でマウスの右ボタンをクリックし、表示されるメニューから[ファイルを追加]を選びますと、[追加するファイルを選択...]ダイアログが表示されます。このダイアログで先ほどコピーした `src` ディレクトリ内にある“`main.c`”、“`sdk_init.c`”、“`nns_util.c`”を選び、[開く]ボタンを押してください。

[開く]ボタンを押しますと、続いて[ファイルの追加]ダイアログが表示されます。サンプルのソースは、[Nitro TS Debug]、[Nitro TS Release]、[Nitro TS ROM]の3つのターゲットに追加しますので、3つのターゲットにチェックが入っていることを確認し、[OK]ボタンを押してください。

図 3-3 ソースファイルの追加画面



## 3.4 NITRO-System をプロジェクトに追加

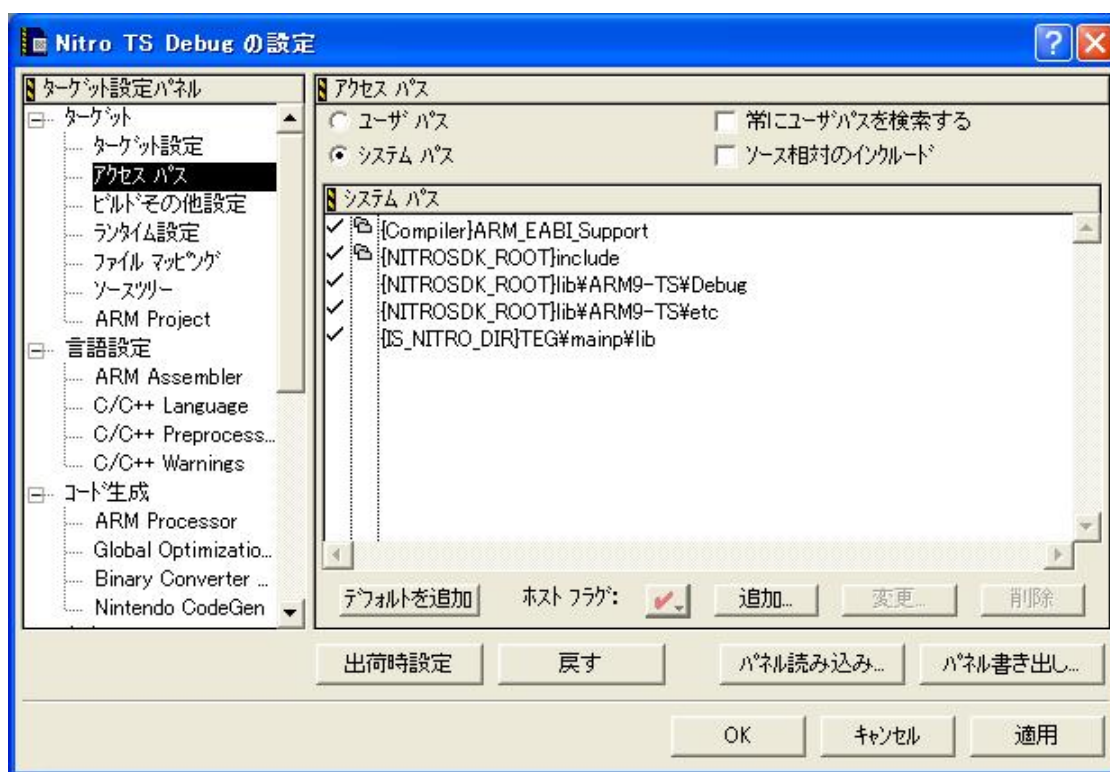
続いて NITRO-System をプロジェクトに追加します。ここでは、[Nitro TS Debug]ターゲットでの設定方法を説明しています。他のターゲットでも設定方法は同じですので、同様に設定してください。

### 3.4.1 デバッグ版のライブラリのアクセスパス設定

プロジェクトのターゲットが[Nitro TS Debug]になっていることを確認し、編集メニューから[Nitro TS Debug の設定]を選択しますと、[Nitro TS Debug の設定]ダイアログが表示されます。

このダイアログのターゲット設定パネルから[ターゲット]→[アクセスパス]を選択しますと、右にアクセスパスのパネルが表示されます。NITRO-System のアクセスパスはシステムパスに追加する必要がありますので、パネル上部の[システム パス]のラジオボタンを選択します。

図 3-4 Nitro TS Debug の設定ダイアログ



このパネル内の[追加...]ボタンを押しますと、[フォルダの参照]ダイアログが表示されます。ここでパスの種類に[NITROSYSTEM\_ROOT]を選んで、NITRO-System のヘッダファイルが収められているディレクトリ(NitroSystem/include)を選択し、[OK]ボタンを押しますと、システムパスに“{NITROSYSTEM\_ROOT}include”が追加されます。



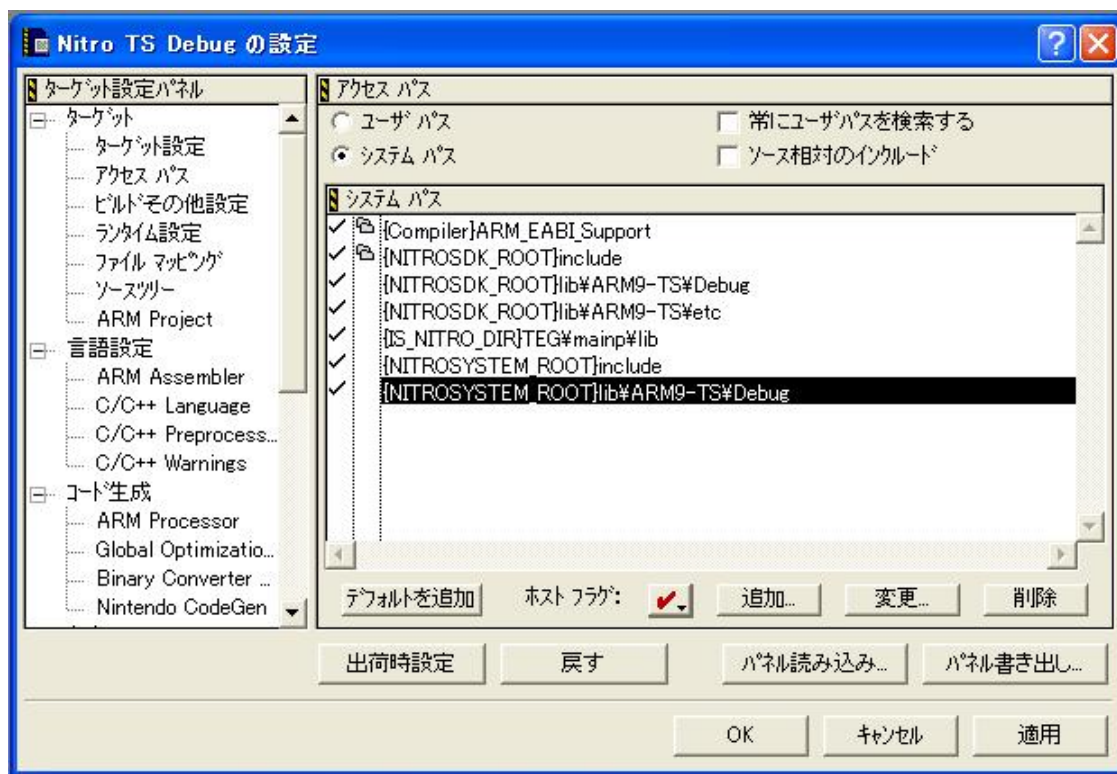
図 3-5 インクルードディレクトリの選択画面



追加されたアクセスパスの頭には、フォルダの形をしたアイコンが表示されています。このアイコンは、設定されてるアクセスパスの下にある全てのディレクトリにパスが通っていることを意味しています。コマンドラインでのビルド環境では、`include` ディレクトリにのみパスを通すように設定されていますので、これと厳密に環境を合わせたい場合には、アイコンをマウスでクリックして消します。

同様に、デバッグ版ライブラリが格納されているディレクトリ `NitroSystem/lib/ARM9-TS/Debug` をシステムパスに追加します。

図 3-6 Nitro TS Debug ターゲットのアクセスパス設定画面



### 3.4.2 デバッグ版のライブラリの追加

---

NITRO-SDK のライブラリが追加されている **TS Libs** グループと同じ構成のグループを作成し、そのグループ内に **NitroSystem** ライブラリを追加することになります。

プロジェクトメニューから[グループ作成]を選びますと、グループ作成ダイアログが表示されます。このダイアログに「**NitroSystem**」と入力して[OK]ボタンを押しますと、プロジェクトファイル一覧に[**NitroSystem**]という名前のグループが作成されます。同様に、[**NitroSystem**]グループ内に、[**TS Debug**]グループを作成します。

[**NitroSystem**]/[**TS Debug**]グループの上でマウスの右ボタンをクリックしますとメニューが表示されますので、このメニューから[ファイルを追加]を選び、**NitroSystem/lib/TS/Debug** 内の下記の6つのライブラリを追加します。

```
libnnsnd.a  
libnnsfnd.a  
libnns2d.a  
libnns3d.a  
libnnsbfd.a  
libnnsmdcs.a
```

なお、このときに表示される[ファイルの追加]ダイアログでは、[**Nitro TS Debug**]ターゲットだけにチェックを入れて[OK]ボタンを押してください。

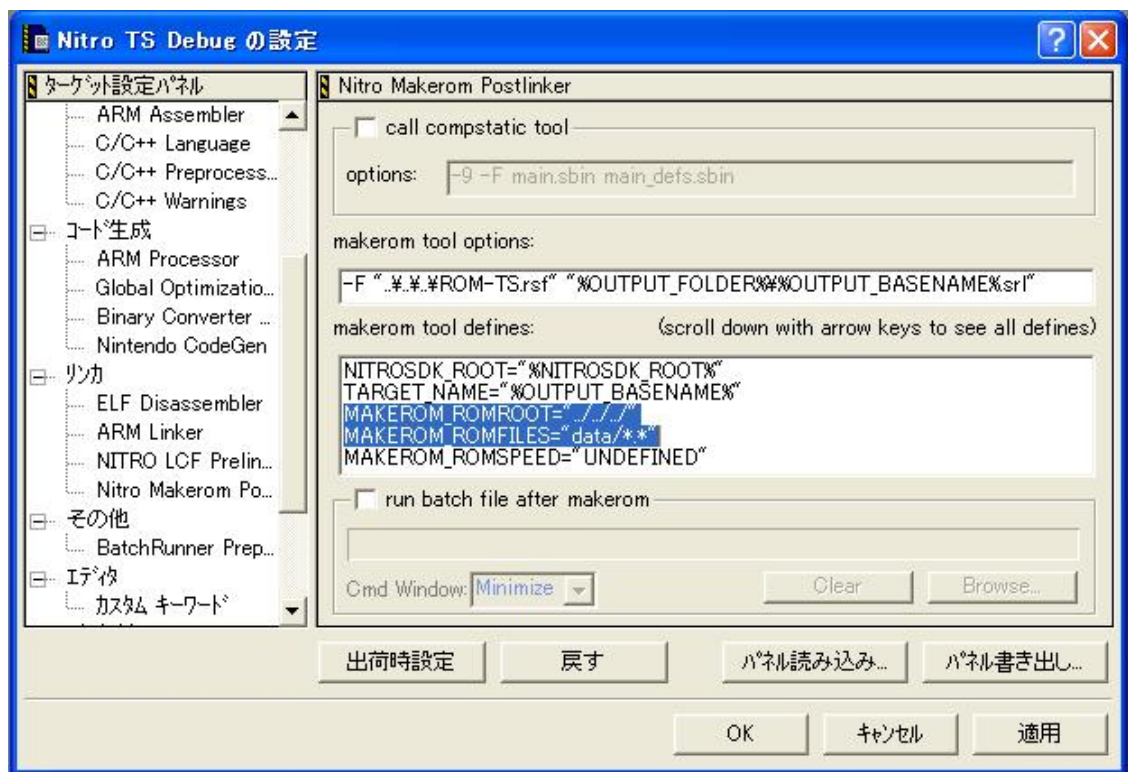
## 3.5 サンプルのデータをプロジェクトに追加

---

サンプルプログラムが使用するデータをプロジェクトに登録します。

プロジェクトのターゲットが[**Nitro TS Debug**]になっているのを確認し、編集メニューから[**Nitro TS Debug** の設定]を選択します。ターゲット設定パネルから[リンカ] - [**Nitro Makerom Postlinker**]を選択しますと、[**Nitro Makerom Postlinker**]パネルが表示されます。

図 3-7 Nitro TS Debug ターゲットの Postlinker 設定画面



このパネルの[makerom tool defines]の中に表示される MAKEROM\_ROMROOT と MAKEROM\_ROMFILES の設定内容を下記の設定に変更します。

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

MAKEROM\_ROMROOT には、データが存在するディレクトリを、“srl”ファイルが出力されるディレクトリからの相対パスで設定します。また、MAKEROM\_ROMFILES には、ROM に含めるファイルを指定します。

### 3.6 プリコンパイルヘッダ使用の準備

プロジェクトステーションナリ ARM9-TS Application からプロジェクトを作成すると、デフォルトの状態ではプリコンパイルヘッダを使用する構成となっています。プロジェクトには、C 言語用と C++言語用のふたつのプリコンパイルヘッダファイルが登録されています。

このプリコンパイルヘッダファイルの Nitro\_TS.pch (C 言語用) と Nitro\_TS.pch++ (C++言語用) に、NITRO-System のヘッダファイルを追記します。

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

nnsys\_prefix.h は、make のビルド環境でコンパイラに渡されるマクロを定義したヘッダファイルです。このヘッダファイルをインクルードすることにより、make でのビルドと同じ設定でコンパイルすることができます。

また、プリコンパイルヘッダを使用する場合には、サンプルプログラムのヘッダファイルやソースの中で、NITRO-System のヘッダファイルをインクルードする必要はありませんので、削除します。

### 3.7 プリコンパイルヘッダを使用しない場合

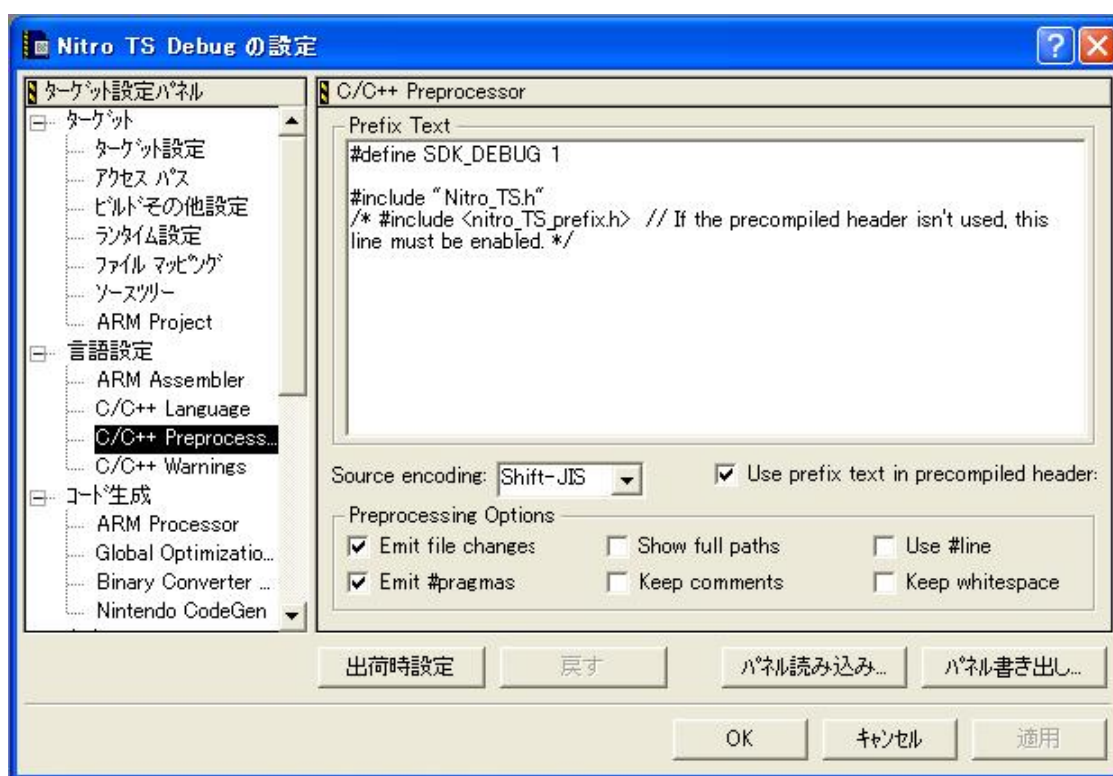
プリコンパイルヘッダを使用しない構成に変更するには、プロジェクトから下記の3つのファイルを削除します。

- Nitro\_TS.pch
- Nitro\_TS.h
- Nitro\_TS.pch++

その後、ターゲット設定パネルから[言語設定] - [C/C++ Preprocessor]の Prefix Text の内容について、下記の変更を行います。

- 「#include "Nitro\_TS.h"」の行を削除します。
- 「#include <nitro\_TS\_prefix.h>」の行のコメントアウトを解除し、このインクルードを有効にします。
- 「#include <nnsys\_prefix.h>」を「#include <nitro\_TS\_prefix.h>」の下に追加します。

図 3-8 Nitro TS Debug ターゲットの C/C++ Preprocessor 設定画面



### 3.8 プロジェクトのビルド

これで Archive サンプルのビルドの準備が完了しました。プロジェクトのターゲットが[Nitro TS Debug]になっていることを確認し、プロジェクトメニューから[メイク]を選択して(または F7 キーを押して)、ターゲットをビルドしてください。

[Nitro TS Release]ターゲットと[Nitro TS ROM]ターゲットに関しても、設定方法は[Nitro TS Debug]ターゲットと同じです。同様にプロジェクトの設定を行い、ビルドを行ってみてください。

## 4 MultiCellAnimation サンプルのビルド

この章では、NITRO-System の g2d ライブラリの MultiCellAnimation サンプルをビルドするまでの手順を説明します。こちらはプライベートなライブラリを使用している場合の例となります。

### 4.1 g2d デモライブラリのビルド

g2d ライブラリのサンプルは、“libnns2d\_demo.a”という g2d サンプル専用のプライベートなデモライブラリを使用しています。まず最初に、このデモライブラリのプロジェクトを作成します。

#### 4.1.1 デモライブラリプロジェクトの作成

CodeWarrior IDE のファイルメニューから[新規]を選択しますと、新規ダイアログが表示されます。この新規ダイアログのプロジェクトタブの中での[NITRO ステーションナリ]を選択し、右にプロジェクト名と場所を入力して[OK]ボタンを押してください。

[OK]ボタンを押します新規プロジェクトダイアログが表示されます。このダイアログのプロジェクトステーションナリで[SDK3.0]－[ARM9-TS Library]－[C]を選択し、[OK]ボタンを押すとライブラリ用のプロジェクトが作成されます。

図 4-1 ライブラリステーションナリの選択画面



#### 4.1.2 サンプルのソースとデータの準備

デモライブラリ用のプロジェクトができましたので、g2d サンプルのデモライブラリのソースを準備します。

まず最初に、ステーションナリにより予めプロジェクトに追加されている“main.c”は使用しませんので削除してください。次に NITRO-System の NitroSystem/build/demos/g2d/demolib ディレクトリから“include”と“src”の2つのディレクトリをプロジェクトのディレクトリにコピーします。

### 4.1.3 デモライブラリのソースをプロジェクトに追加

デモライブラリのソースをプロジェクト内の[Sources]グループ内に追加します。ソースの追加方法は、Archive サンプルの場合と同じです。

まずステーションナリで追加されている“main.c”は必要ありませんので、ファイルをプロジェクトから削除してください。次に、先ほどプロジェクトディレクトリにコピーした src ディレクトリ内にある“fontData.c”、“loader.c”、“print.c”、“system.c”の4つのファイルをプロジェクトに追加してください。このとき表示される[ファイルの追加]ダイアログでは、3つのターゲット全てにチェックを入れることを忘れないでください。

### 4.1.4 NITRO-System をプロジェクトに追加

NITRO-System をプロジェクトに追加します。ここでは、[Nitro TS Debug]ターゲットでの設定方法を説明しています。他のターゲットでも設定方法は同じですので、同様に設定してください。

デモライブラリのビルドでは、NitroSystem のライブラリのバイナリ(.a ファイル)を追加する必要はありません。NitroSystem の include ディレクトリへのアクセスパスの設定だけを行います。設定方法は、Archive サンプルの場合と同じです。

[Nitro TS Debug の設定]ダイアログのターゲット設定パネルから[ターゲット]－[アクセスパス]を選択し、システムパスに NITRO-System のヘッダファイルが収められているディレクトリ(NitroSystem/include)を追加してください。

### 4.1.5 ライブラリのビルド

これでデモライブラリのビルドの準備が完了しました。プロジェクトのターゲットが[Nitro TS Debug]になっていることを確認し、プロジェクトメニューから[メイク]を選択して(または F7 キーを押して)、ターゲットをビルドしてください。

[Nitro TS Release]ターゲットと[Nitro TS ROM]ターゲットに関しても同様の設定を行って、ビルドしてください。

## 4.2 MultiCellAnimation サンプルのビルド

g2d ライブラリのサンプルである MultiCellAnimation サンプルのビルドについて説明します。

### 4.2.1 MultiCellAnimation サンプルのプロジェクト作成

MultiCellAnimation サンプルのプロジェクトの作成方法は、Archive サンプルのプロジェクトの作成方法と同じです。Archive サンプルのプロジェクト作成方法に従ってプロジェクトを作成し、サンプルのソースとデータ、及び NITRO-System のライブラリを追加してください。

### 4.2.2 デモライブラリの追加

g2d ライブラリのサンプルでは g2d のデモライブラリを使用していますので、デモライブラリもプロジェクトに追加する必要があります。ここでは、[Nitro TS Debug]ターゲットでの設定方法を説明しています。他のターゲットでも設定方法は同じですので、同様に設定してください。

#### 4.2.2.1 デモライブラリのアクセスパスの設定

デモライブラリはユーザアクセスパスに登録します。[Nitro TS Debug の設定]ダイアログのターゲット設定パネルから[ターゲット]－[アクセスパス]を選択し、ユーザパスに g2d デモライブラリの include ディレクトリとデバッグ版ライブラリが格納されているディレクトリ(デモライブラリプロジェクト内の bin/ARM9-TS/Debug)を追加してください。



#### 4.2.2.2 デバッグ版デモライブラリの登録

NITRO-SDK のライブラリが追加されている[TS Libs]グループと同じ構成のグループを作成し、そのグループ内に g2d デモライブラリを追加することになります。

プロジェクトメニューから[グループ作成]を選びますと、グループ作成ダイアログが表示されます。このダイアログに「G2dDemoLib」と入力して[OK]ボタンを押しますと、プロジェクトファイル一覧に“G2dDemoLib”という名前のグループが作成されます。同様にして、[G2dDemoLib]グループ内に[TS Debug]グループを作成します。

[G2dDemoLib]/[TS Debug]グループの上でマウスの右ボタンをクリックしますとメニューが表示されますので、このメニューから[ファイルを追加]を選び、デモライブラリのプロジェクトディレクトリ内にあります bin/ARM9-TS/Debug/libg2d\_demo.a ファイルを追加します。

なお、このとき表示されるファイルの追加ダイアログでは、[Nitro TS Debug]ターゲットだけにチェックを入れて[OK]ボタンを押してください。

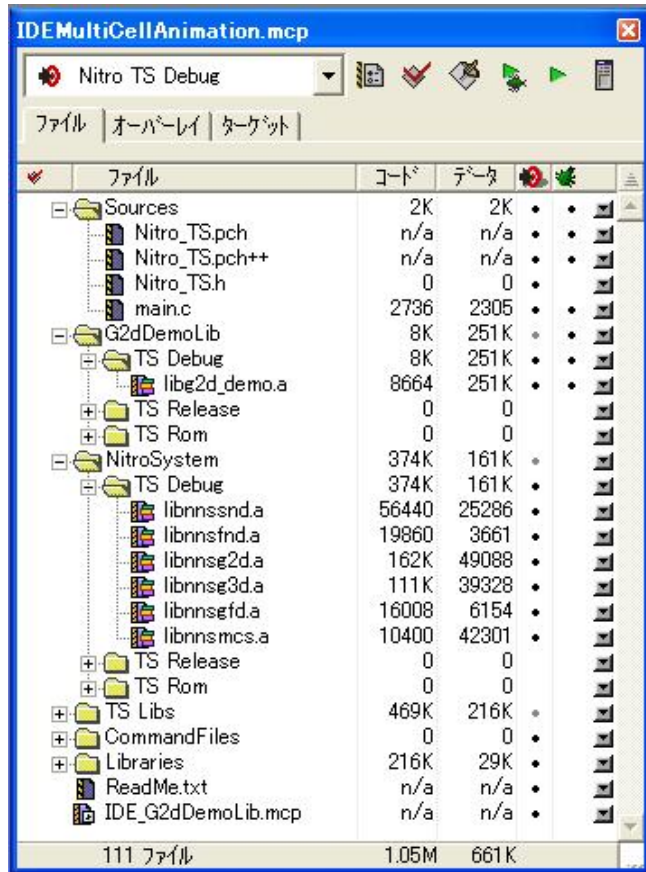
#### 4.2.3 プロジェクトのビルド

これで MultiCellAnimation サンプルのビルドの準備が完了しました。プロジェクトのターゲットが[Nitro TS Debug]になっているのを確認し、プロジェクトメニューから[メイク]を選択して(または F7 キーを押して)、ターゲットをビルドしてください。

[Nitro TS Release]ターゲットと[Nitro TS ROM]ターゲットに関しても同様の設定を行って、ビルドしてください。

MultiCellAnimationプロジェクトウィンドウの内容を図 4-2 に示します。

図 4-2 MultiCellAnimation プロジェクトの内容



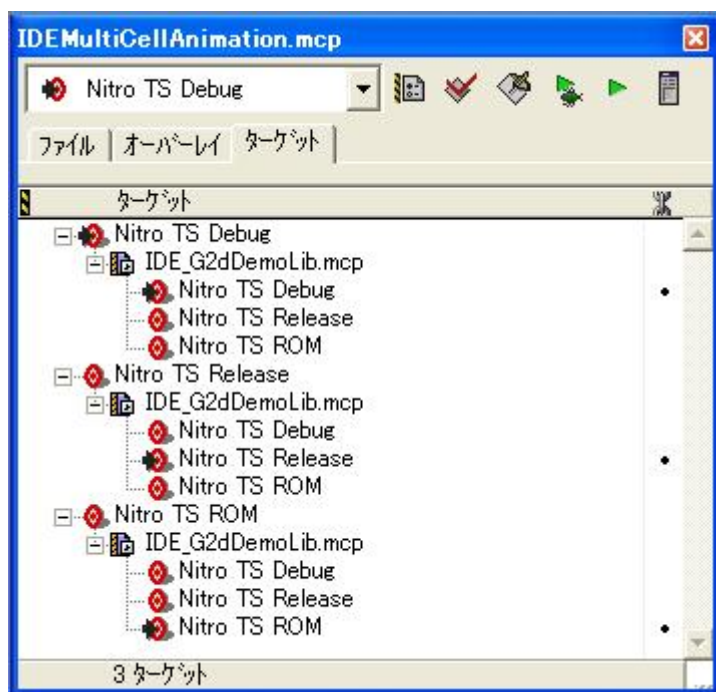
## 4.3 サブプロジェクト

図 4-2 では、MultiCellAnimationのプロジェクトに、g2dデモライブラリのプロジェクトをサブプロジェクトとして追加しています。この様にすれば、MultiCellAnimationのプロジェクトをビルドするときに、g2dデモライブラリのビルドが必要であれば、先にデモライブラリがビルドされます。

g2d デモライブラリプロジェクトの追加は、プロジェクトへのファイルの追加と同じ方法で可能です。連携してビルドを行うためには、追加した後にターゲットの設定を行う必要があります。ターゲットの設定は、プロジェクトのターゲットタブで設定します。

ターゲットタブでは、サンプルの各ターゲットをビルドする場合に、デモライブラリのどのターゲットをビルドするかを指定します。通常、サンプルの[Nitro TS Debug]ターゲットをビルドする場合には、デモライブラリの[Nitro TS Debug]ターゲットをビルドすれば良いので、デモライブラリの[Nitro TS Debug]ターゲットの前にある“的”のアイコンをクリックして、矢印を付けます。

図 4-3 ターゲットの設定





CodeWarrior は、フリースケール社の商標です。

その他、記載されている会社名、製品名等は、各社の登録商標または商標です。

© 2007 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。