TWL-SDK

NAND Application Development Manual

Version 2009/05/22

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

TWL-06-0008-001-C
Released: August 7, 2009

Table of Contents

Code

Tables

Revision History

Revision Date	Description
2009/05/22	In section 4.3 Creating an RSF, noted that an application's size will increase by 16 KB when it uses a sub-banner. Added a description of the TCL library to section 10.5 Photo Database.
2009/01/20	In Chapter 9 Copying to an SD Card, added text explaining that NAND applications cannot be copied or written back on test units..
2008/12/17	Added a note to section 6.2 Simple Debugging.
2008/11/06	Added a link to the *Function Reference Manual*. Revised "DS Menu" to "DSi Menu." Added mention of the properties that are prohibited to change during version updates.
2008/10/23	Deleted Chapter 11.
2008/09/30	Changed "GameCode" to "InitialCode" in section 4.3.
2008/09/16	Initial version.

1 Introduction

This document provides technical information required for the development of NAND applications.

This document compares NAND applications with card applications. It is assumed the intended audience has the knowledge required to create card applications.

2 Terminology

The following are brief descriptions of the terminology used in this document. Some of these descriptions may differ from commonly known meanings.

Table 2-1 Terminology Descriptions

Terminology	Description
System memory	The TWL system's internal NAND flash memory.
Card application	TWL applications stored in and run from Game Cards. Data is loaded from the Game Card at run time.
NAND application	TWL applications stored in and run from the system memory. Data is loaded from the system memory at run time.
SRL	TWL application executable file format, files that contain this format, and the file extension.
ROM	Same definition as SRL. However, more emphasis is on data compared to SRL.
RSF (ROM Spec File)	File that contains the parameters used when creating SRL.
Remaster version	SRL version number stored in SRL and coded in RSF.
Company code	An identifier assigned to each publisher that consists of two ASCII characters.
Game code	A unique identifier assigned to each TWL application. Consists of four ASCII characters.
Import	Stores the NAND application in the system memory and sets it to executable.
TAD	The file format used when importing a NAND application to the system memory. SRL is included.
DSi Menu	The screen displayed when TWL starts. Use to select the application you want to run.
Archive	Each file system that is managed by the TWL-SDK FS library. Corresponds to a Windows computer "drive."
ROM archive	Archive that indicates the standard ROM file system provided in TWL-SDK. It is loaded by default into the FS library.
Game banners	Collections of icons, application names, and so on, used to define an application. These are used when a list of applications is displayed in the DSi Menu.
Sub-banners	These are subsets of the game banners. This data is used to swap out image data contained in game banners. Alternately, the mechanism for swapping out banner images.
Save data	Data saved and maintained by an application to be reused after the application closes.
Save data region	A region allocated in the system memory that holds NAND application save data.
Public save data	One of the save data regions. Users can copy this region to an SD card.
Private save data	One of the save data regions. Users cannot copy data from this region to an SD card.
Reference	Documentation under the TWL-SDK `man` directory that describes the APIs and tools.

3 NAND Application and SDK

3.1 NAND Application

A NAND application is an application that can be installed in the TWL internal system memory.

Basic technical differences as compared to card applications are listed below.

- Run from the system memory

  A DS card is not required to run a NAND application. However, you must import the application to the system memory.

- ROM access speed

  Because ROM is in the system memory, the access speed will be different for the ROM archive from FS library. Also, there are restrictions in the access frequency.

- Save data storage location

  The save data is stored in the system memory. For details, see Chapter 7 Save Data.

- Copying to SD Card

  Users can copy applications and save data to an SD Card. For details, see Chapter 9 Copying to an SD Card.

3.2 SDK

The following SDK items are required for NAND applications, but are not required for card applications.

Table 3-1 SDK Items Required for NAND Applications

Item	Description
Libraries	
`include/twl/na.h`	NA library headers. The NA library contains an additional API that can be used for NAND applications. In the case of NAND applications, you must link the NA library because it contains functions that the SDK requires internally for NAND applications.
`include/twl/specfiles/ROM-TS.nand.rsf`	RSF file sample that contains additional items that can be used with NAND applications.
`lib/ARM9-TS/*/libna.TWL*.a`	NA library `.a` files.

Item	Description
Tools	
tools/bin/maketad.exe	A Windows tool that converts SRL to TAD. You must convert to TAD to import a NAND application to the system memory.
bin/ARM9-TS/Rom/TwlNmenu.srl	A TWL tool that imports a NAND application from a TAD file in an SD Card to a TWL development machine.
bin/ARM9-TS/Rom/NandFiler.srl	A TWL tool that can refer to the NAND application's save data region.
bin/ARM9-TS/Rom/TwlNmenu.tad bin/ARM9-TS/Rom/NandFiler.tad	A package created as a NAND application that contains TwlNmenu and NandFiler.
Documents	
docs/README/NandAppManual.pdf	This document.
man/en_US/na/list_na.html	NA library reference manual.
man/en_US/demos/nandApp/nandAppdemos.html	Reference manual for NAND application sample demos.
man/en_US/romfiles/TwlNmenu.html	Reference manual for TwlNmenu.
man/en_US/romfiles/NandFiler.html	Reference manual for NandFiler.
man/en_US/tools/maketad.html	Reference manual for maketad.
man/en_US/tools/SaveDataSize.html	A list of values that can be used for save data region size.
Sample Demos	
build/demos.TWL/nandApp	NAND application sample demos.

4 Creating NAND Applications

To run a NAND application on a TWL system, you must first create a TAD file.

To create a TAD file:

1. Create a program in the same way that you would create a card application.
2. Link, including the NA library.
3. Create an RSF specific to the application.
4. Use the application-specific RSF to create an SRL.
5. Create a TAD from the SRL.

The following sections detail each of these procedure steps.

4.1 Creating a Program

Basically, this is the same as for a card application. However, note that the following items are different.

- The access speed to ROM archive using the FS library is different, and the access frequency is restricted.
- Save data is saved in `dataPub` and/or `dataPrv` archives. See Chapter 7 Save Data.
- The application alone may be updated while keeping the save data as is. See Chapter 8 Versions.
- Users can clear the `dataPrv` archive. See Chapter 9 Copying to an SD Card.

4.2 Linking the NA Library

In the case of NAND applications, when you link a program, you must link the NA library (`libna.TWL*.a`).Using the API contained in the NA library is optional. However, even if you do not use the API, you must link the NA library since the NA library contains functions that are required to operate NAND applications.

4.3 Creating an RSF

You must prepare an RSF (ROM Spec File) to create an SRL. In TWL-SDK, an `AppendProperty` section has been added to RSF. This specifies parameters related to NAND applications. When creating a NAND application, you must create an SRL using an RSF in which the `AppendProperty` section contains the appropriate parameters. If the appropriate parameters are not set, `maketad.exe` reports an error when you create the TAD. Create an appropriate RSF for each NAND application.

The following sample RSF contains possible settings for NAND applications: `include/twl/specfiles/ROM-TS_nand.rsf`.

Here is a brief description of the settings in the AppendProperty section. For details and RSF specifications other than those found in the AppendProperty section, see the makerom.TWL reference.

- Media

  Specifies the ROM storage media. The default is GameCard. You must specify NAND for a NAND application.

- InitialCode

  Specifies the game code. For each application, allocate a game code consisting of 4 ASCII characters. NAND applications imported to the system memory are identified by their game code. Note that if a NAND application with the same game code has already been imported, it will be overwritten.

- PublicSaveDataSize

  Specifies the size of the public save data region. This save data region can be copied to an SD card. For information on the save data region, see Chapter 7 Save Data.

  For information on values that can be specified, see **Tools → Tools Related to ROM Images → Save Data Size List** in the *Function Reference Manual*.

- PrivateSaveDataSize

  Specifies the size of the private save data region. This save data region cannot be copied to an SD card. For information on the save data region, see Chapter 7 Save Data.

  Specifiable values are the same as for PublicSaveDataSize.

- SubBannerFile

  Specifies whether to use sub-banners. The default is FALSE. Set TRUE to use sub-banners; otherwise, set FALSE. See section 10.1 Sub-Banners for information on sub-banners.

  A NAND application's overall size increases by 16 KB if it uses a sub-banner. For details on calculating the overall size of a NAND application, see **Built Programs → NAND Application-Related → TwlNmenu.srl** in the *Function Reference Manual*.

4.4 Creating an SRL

SRLs are made the same way as with card applications. However, note that the RSF that you use for the SRL must conform to section 4.3 Creating an RSF.

4.5 Creating a TAD

To create a TAD from an SRL, enter the SRL file path as a command-line argument and run maketad.exe. The default output is a filename consisting of the RSF TitleName property value with the extension .tad. You can also specify the output filename with the -o option. For more details, see the maketad reference.

Example:

```
maketad.exe MyNandApp.srl –o MyNandApp.tad
```

4.6 Using the TWL-SDK Build System

The TWL-SDK make build system provides rules for creating an SRL. Rules are also provided for creating NAND applications. To enable these rules, in Makefile, set the TWL_NANDAPP variable to TRUE. The rules for creating NAND applications include rules for creating a TAD and the setting to add the NA library to the link target. Using the rules makes it easy to create TADs.

When you use the TWL-SDK build system to create a NAND application, change the following three areas in card application Makefiles.

- Set TWL_NANDAPP to TRUE.
- Set the path to the RSF for the NAND application in ROM_SPEC.
- Specify a filename with the extension .tad (for example, main.tad) in TARGET_BIN.

Run make to create the TAD. For detailed examples of Makefile, see the NAND application sample demos.

Code 4-1 Sample Makefile

```
...Snip...
TARGET_PLATFORM := TWL
TWL_ARCHGEN    := LIMITED
TWL_NANDAPP    =  TRUE

SRCS          =   main.c
TARGET_BIN     =   MyNandApp.tad

ROM_SPEC       =   MyNandApp.rsf

include $(TWLSDK_ROOT)/build/buildtools/commondefs

do-build:      $(TARGETS)

include $(TWLSDK_ROOT)/build/buildtools/modulerules
...Snip...
```

5 Import and Run

To run a NAND application, you must first import it into TWL system memory. Use TWL Nmenu or IS-TWL-DEBUGGER to import NAND applications.

5.1 Import Precautions

5.1.1 Game Code

NAND applications imported to the system memory are identified by their game code. If a NAND application having the same game code as an existing application is imported, the import will overwrite the existing application.

5.1.2 Versions

All NAND applications have versions. In the case of an overwrite import, the version you are importing must be either the same or newer than the version that has already been imported. Importing a previous version fails.

If you must import a previous version, first remove the already imported NAND application and then import the previous version. You can remove NAND applications using the TWL Nmenu.

See Chapter 8 Versions for information on NAND application versions.

5.1.3 Save Data Region and Sub-Banners

In the case of overwrite import, normal save data and sub-banner content are retained. However, all save data region and sub-banner content is cleared if even one setting related to save data region size or sub-banners enabled/disabled differs between the application being imported and the one that has already been imported.

5.2 Using TWL Nmenu

The steps for importing and running a TAD are shown below.

1. Save the TAD file in an SD Card and insert it in the TWL development machine's SD Card slot.
2. Run `TwlSDK/bin/ARM9-TS/Rom/TwlNmenu.srl` from the development machine.
3. Use TWL Nmenu on the development machine to import the TAD.
   (a) Use right and left on the +Control Pad on the development machine to switch to SD Mode.
   (b) Select the TAD file and press the A Button. Use up and down on the +Control Pad to select directories or files in directories. To move to a directory, select the directory and press the A Button.
   (c) An import confirmation screen appears. Press the A Button.
   (d) Wait for importing to complete.

4. Use TWL Nmenu on the development machine to run the NAND application.
   (a) Use right and left on the +Control Pad on the development machine to switch to NAND Mode.
   (b) A list of installed NAND application game codes is displayed in NAND Mode. Use up and down on the +Control Pad to select the NAND application you want to run, and then press START.
   (c) The NAND application starts.

5.3 Using IS-TWL-DEBUGGER

To import a TAD, either drag it into the IS-TWL-DEBUGGER software window or click **File**, click **Open**, select the TAD, and then click **OK**. After that, you can debug a NAND application the same way you debug a card application.

6 Debug

Use IS-TWL-DEBUGGER to debug.

6.1 Normal Debugging

Use the method in section 5.3 Using IS-TWL-DEBUGGER for normal debugging. We strongly recommend this method for debugging.

6.2 Simple Debugging

Simple debugging using SRL or TLF is available once the NAND application that you want to debug is imported. As with card applications, you can open an SRL or a TLF from IS-TWL-DEBUGGER software.

Note the following differences between normal debugging and this method.

- ROM archive does not access the system save memory. Instead it accesses a DS Game Card (or equivalent).

  In this case, the same state as the one entered when starting a card application is entered, and the access speed to the ROM archives is halved compared to importing and running a NAND application. Because the component load is lightened, wireless, microphone, and sound operation may be different.

- Save data and sub-banners are created when the TAD file is imported, so you must always import NAND applications once. In addition, you must re-import to apply changes to settings, such as the save data region size or whether sub-banners are enabled.

In the last stages of development, be sure to import and verify operations using the method described in section 6.1 Normal Debugging.

7 Save Data

Unlike card applications, NAND applications cannot use an on-card backup device. Instead, a save data region is allocated in the system memory for each NAND application.

7.1 Using the Save Data Region

The NAND application save data region is an independent file system, which is loaded as part of the FS library archive. When a NAND application calls the `FS_Init` function, the `dataPub` and `dataPrv` archives are loaded internally. Prepend these archive names to the path when using the FS library to create files and directories, and to read and write files in the save data region.

The archive `dataPub` is the save data region that is copied along to an SD Card when copying a NAND application to an SD Card. Store the normal save data that is safe for copying in this region. This region is called "public save data."

The archive `dataPrv` is a save data region that is never copied. Store the save data you do not want copied in this region. This region is called "private save data."

7.2 Save Data Region Size

7.2.1 Size Specification

A save data region size is selected for each NAND application and must be specified in RSF. For information on the save data region size specification, see section 4.3 Creating an RSF.

The save data region size is set to 0 by default. If the save data region size is 0, the corresponding archives will not load.

7.2.2 Usable Size

The save data region also contains a filing system management region and other items. Therefore, the region that the application can actually use is smaller than the specified value. For information on the save data region size and the usable size, see **Tools → Tools Related to ROM Images → Save Data Size List** in the *Function Reference Manual*.

You can create any files and directories in the save data region as long as you do not exceed the usable size. If you attempt to create files or directories that exceed the size, the FS API will return an error.

Use the `FS_GetArchiveResource` function to get the maximum usable size and remaining space.

7.2.3 Number of Files That Can Be Created in the Root Directory

There is no limit to the number of files that can be created in the save data region. However, there is a limit to the number of files and directories that can be created in the save data region's root

directory. The number of files and directories that can be created in the root directory is determined by the lengths of the filenames and directory names.

The number of files that can be created in the root directory depends on the size of the save data region. For a list of values, see **Tools → Tools Related to ROM Images → Save Data Size List** in the *Function Reference Manual*.

7.2.4 Cluster Size

When a file is created in the save data region, a file of *n* bytes consumes a region not of *n* bytes, but of *n* rounded up to a multiple of a certain value. This certain value is the cluster size. For example, if the cluster size is 512 bytes, both a 1-byte and a 512-byte file consume a region of 512 bytes each. Likewise, both a 513-byte file and a 1024-byte file consume a region of 1024 bytes.

The cluster size depends on the save data region size. For a list of values, see **Tools → Tools Related to ROM Images → Save Data Size List** in the *Function Reference Manual*.

7.3 SD Card Write Back

Private save data is not copied to SD Cards. Therefore, when you write back a NAND application that has been copied to an SD Card, private save data is cleared. Applications must assume that only private save data is always cleared.

For details on copying to SD Cards, see Chapter 9 Copying to an SD Card.

8 Versions

NAND applications have versions.

8.1 Version Format and Specification

NAND application versions consist of 0-255 major versions and 0-255 minor versions, connected by a period. For example, major version 2, minor version 1 would be 2.1.

The SRL remaster version is used as the NAND application's major version. If the SRL remaster version is 1, the NAND application major version is 1. After the NAND application is released, the SRL version is incremented by 1 for each version upgrade.

The NAND application minor version is reserved, and currently is always set to 0.

8.2 Version Upgrades

Unlike card applications, with NAND applications it is possible to upgrade software that the user currently owns. When a version is upgraded, the save data is retained. You must implement so that the new version of the NAND application can appropriately handle save data for all previous versions. To do so, either permanently maintain the save data format or embed a version number so that the program can handle the changes.

It is prohibited to change any of the following RSF properties in a version update.

- `TitleName`
- `MakerCode`
- `CardRegion`
- `InitialCode`
- `PublicSaveDataSize`
- `PrivateSaveDataSize`
- `SubBannerFile`

8.3 Version Downgrades

If a user first copies the old version of a NAND application to an SD Card, he or she is able to downgrade at any time by writing back.

When a NAND application is copied to an SD Card, the save data is also copied; when writing back to the system memory, the save data is also written back. Therefore, you will not have the problem of the NAND application being old and the save data being new. However, note that when you write back from an SD Card to the system memory, although the public save data is written back from the SD Card, the private save data is cleared.

9 Copying to an SD Card

In the TWL system settings, the Software Management screen provides a feature that allows a user to copy NAND applications to an SD Card. For reference, the following provides information on copying to an SD Card.

However, the various test unit system menus updated with SystemUpdater cannot perform the NAND application copy or write-back operations (described below) on the Software Management screen.

9.1 Copy

When a NAND application is copied to an SD Card, the public save data is copied at the same time as the application. Conversely, the private save data is not copied. The NAND application and the public save data are copied to the SD Card as a single file.

Data copied to the SD Card is encrypted and tamper-proofed.

9.2 Writing Back a Copy

When a copy of a NAND application on an SD Card is written back to a TWL system, the public save data is written back along with the NAND application. The public save data on the TWL system is always overwritten by the data on the SD Card. This means that the user could roll back save data.

If you want to prevent rolling back save data, you can use private save data. However, because the private save data is not copied to an SD Card, when a user writes back to the system memory, the private save data is cleared. Applications must assume that there is always a possibility that they can be starting up with only the private save data cleared. You must assure that the behavior will be normal, even if public save data is maintained and private save data is cleared.

Consider these characteristics and determine: of the data that the application should save, which should be placed in public save data and which should be placed in private save data. Because, as noted above, there is a possibility that private save data will be lost, be careful to place information in private save data only when necessary.

9.3 Limitations on Writing Back

TWL restricts writing back copies from an SD Card, so it is only possible to write back to the TWL system that created the copy. It is not possible to write back to any TWL system other than the one that created the copy. Therefore, it is not possible to copy save data to another system.

Because this restriction is implemented by the system, the application does not need to embed information in save data that will define the system, such as a MAC address, and determine whether the system matches. Because this would cause problems if, for example, a module is exchanged during system repair, such an implementation is prohibited.

10 Additional Features Available for NAND Applications

NAND applications can use a number of additional features not available to card applications. Here is a brief introduction.

10.1 Sub-Banners

NAND applications and card applications both have banners. The DSi Menu lists banners for NAND applications that are installed in system memory. The NAND application starts when the user selects it from the menu.

In the case of card applications, the banner specified when SRL was created is always used, and it was not possible to change banner images after that. In the case of NAND applications, it is possible to use substitute banners called sub-banners. Using this mechanism, it is possible to change banner images.

NAND applications can change sub-banners, making it possible, for example, to change banner images according to game progress.

For more information, see **NA → Overview → Sub-Banners** in the *Function Reference Manual*. The sample demo `SubBanner` is a demo for sub-banners.

10.2 Embedded Fonts

Three types of fonts are embedded in the TWL system memory: large, medium, and small. NAND applications can freely use these fonts.

Embedded fonts are stored in TWL-System NFTR format and support the following character sets.

- ASCII
- ISO 8859-1
- ISO 8859-7
- CP 932
- CP 1252
- CP 1253
- JIS X0201
- JIS X0208
- DS External
- Wii External

For more information, see **NA → Overview → Built-in Fonts** in the *Function Reference Manual*. The sample demo `sharedFont` is a demo that loads embedded fonts.

10.3 Save Data Access by Other NAND Applications

NAND applications can read and write save data from other NAND applications if the applications have the same company code. They can load the save data region from other applications as an FS library archive. Doing this they can manipulate the save data using the FS library.

For more information, see **NA** → **Overview** → **Archives** in the *Function Reference Manual*. The sample demo `other_backup` shows how to read and write the save data of another NAND application (in this case, save data from the `backup` demo).

10.4 Application Reset

NAND applications cannot use `OS_ResetSystem` to reset the applications due to technical restrictions. Instead, `OS_RebootSystem` is prepared. Also, you can use `OS_JumpToSystemMenu` to stop an application's processes and return to the DS Menu.

For more information, see **OS** → **Overview** → **Reset** in the *Function Reference Manual*.

10.5 Photo Database

TWL has a photo database, which is a mechanism by which NAND applications can share JPEG images. Use the DSi Photo Database Library (TCL) to access shared JPEG images. The TCL library allows you to load JPEG images saved in NAND memory by Nintendo DSi Camera (or some other means) and to write JPEG images to a path that can be read by Nintendo DSi Camera.

For details, see **TCL** → **Overview** → **About TCL** in the *Function Reference Manual*.