

N I N T E N D O
NITRO-SampleTools

TextureViewer Manual

Version 1.2.0

The contents in this document are highly
confidential and should be handled accordingly.

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo of America Inc. and/or Nintendo Company Ltd. and are protected by Federal copyright law. 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	Overview	5
2	Procedures: Creating NTF to Displaying on the NITRO LCD	6
3	TextureViewer Operation	10
4	Precautions for Use	11

List of Figures

Figure 2-1	Confirming whether NTF was output.....	6
Figure 2-2	Confirming whether textureData.c and textureData.h are output.....	7
Figure 2-3	Copying NTF andTextureData.c to TextureViewer	7
Figure 2-4	Copying textureData.h to TextureViewer	8
Figure 2-5	Cygwin Execution Screen	8
Figure 2-6	Verifying main.srl	9

List of Tables

Table 3-1	TextureViewer Operation	10
-----------	-------------------------------	----

Revision History

Version	Revision Date	Content
1.2.0	07/26/2004	<ul style="list-style-type: none">• Replaced Figures 2-7 and 2-8.• Updated support for SDK 2.0.<ul style="list-style-type: none">◦ Changed <code>main.bin</code> to <code>main.srl</code>. (p. 9)◦ Changed ARM9-TEG to ARM9-TS. (p. 9)• Changed, revised TextureViewer source code<ul style="list-style-type: none">◦ Changed <code>makefile</code> (<code>.bin</code> -> <code>.srl</code>).◦ Revised <code>mainBg.c</code> (Added <code>DC_FlushRange()</code>).
1.1.0	04/23/2004	Corrected typos. Corrected NITRO header to read NITRO-SampleTools
1.0.0	03/05/2004	Added "palette color 0 transparent color toggle" to Operation
0.5.0	01/30/2004	Initial Release

1 Overview

TextureViewer is a simple texture viewer for verifying texture data for NITRO (called NTF below) on NITRO production units. By following the procedures in this manual, you will be able to easily include NTF in the program, even without knowledge of the program. You will also be able to verify texture image quality on the NITRO liquid crystal display.

TextureViewer uses a system that adjusts the polygon size for the purpose of applying texture according to the texture size. Accordingly, frequent scaling is unnecessary for any size textures and previewing multiple textures on the NITRO liquid crystal display is fast and easy.

2 Procedures: Creating NTF to Displaying on the NITRO LCD

(1) Start ntexconv and create NTF from the image file.

Input the texture format you want to convert with ntexconv in the command line, execute the command, and create NTF. Included in TXLib, ntexconv is a tool that creates NTF from BMP and TGA image files. Be sure to specify "Output data for TextureViewer" and execute. Refer to the ntexconv manual for information on how to use ntexconv.

(2) Confirm that NTF and texture data files (`textureData.c` and `textureData.h`) have been output.

When ntexconv is executed, NTF and the texture data files (`textureData.c` and `textureData.h`) are output. The texture data files are created in the same directory as the script files.

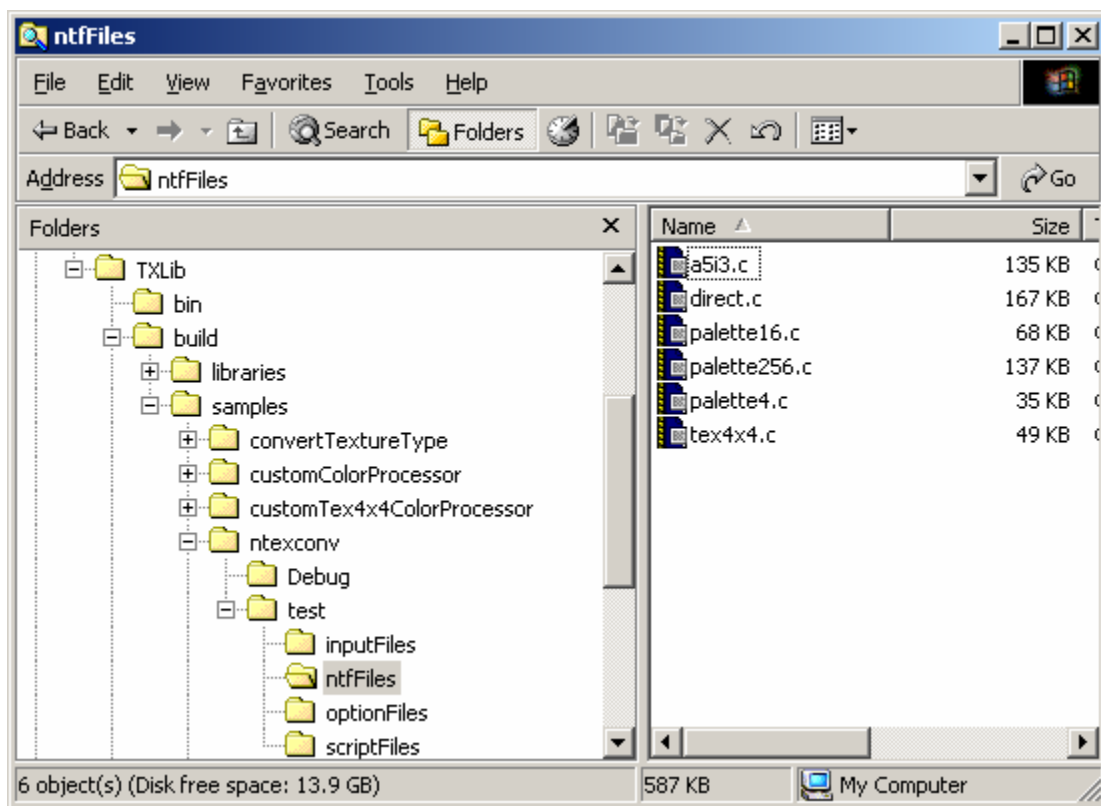


Figure 2-1 Confirming whether NTF was output

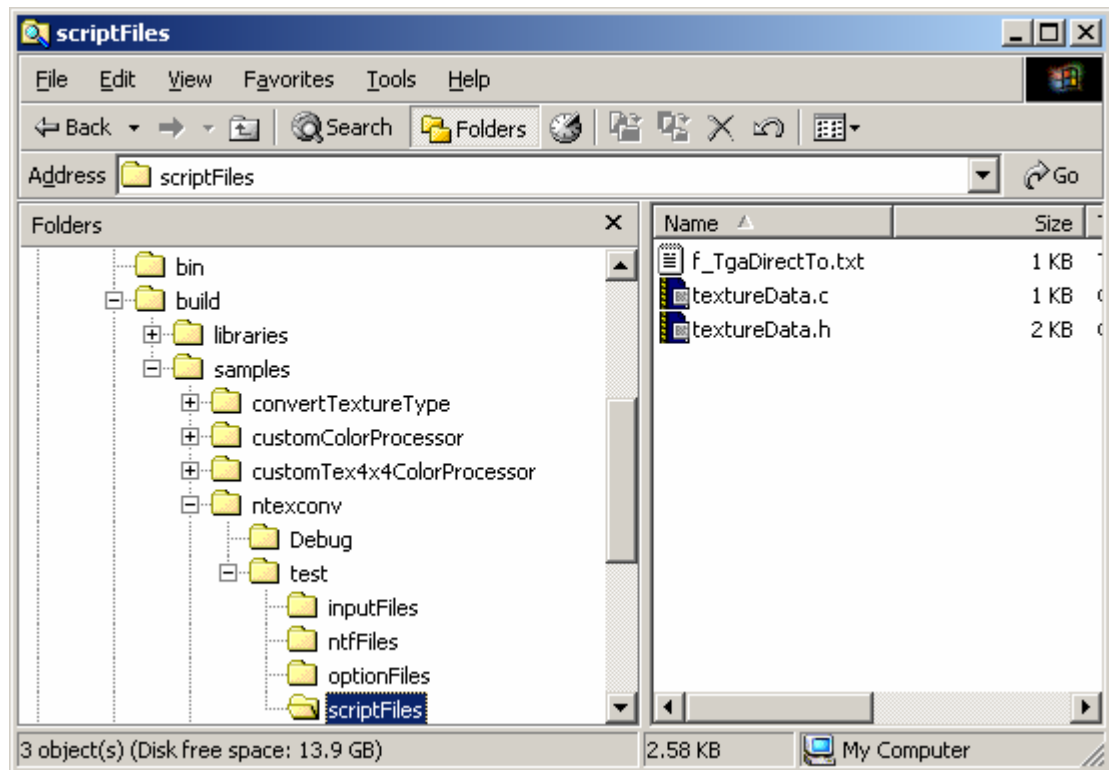


Figure 2-2 Confirming whether textureData.c and textureData.h are output

(3) **Copy NTF and textureData.c to the TextureViewer source file folder.**

Copy NTF and textureData.c to the TextureViewer source file folder
(TextureViewer\src\texture).

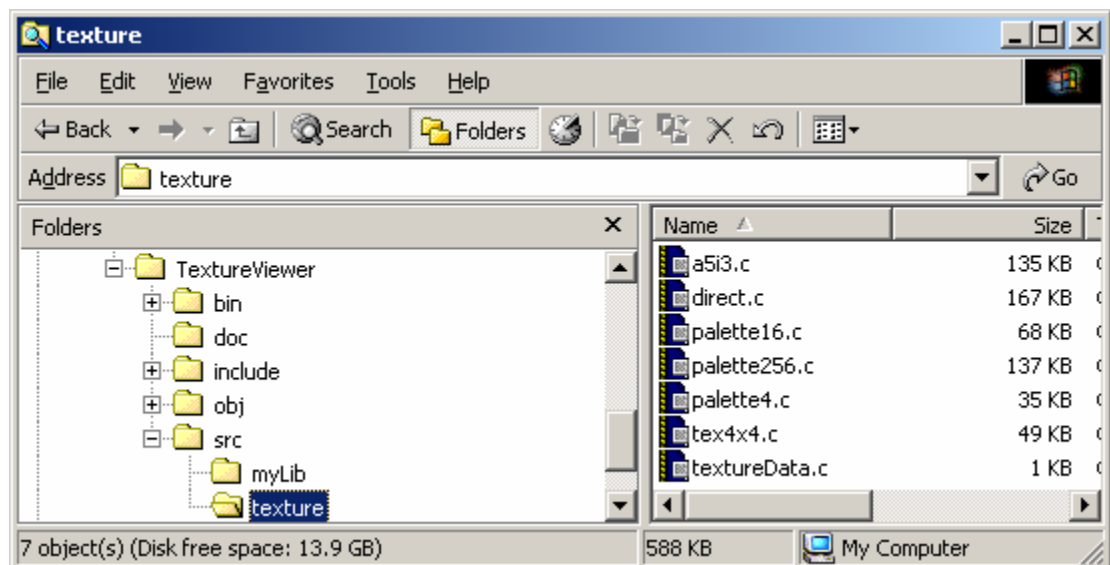


Figure 2-3 Copying NTF and TextureData.c to TextureViewer

(4) **Copy textureData.h to the TextureViewer header file folder.**

Copy textureData.h to the TextureViewer header file folder (TextureViewer\include).

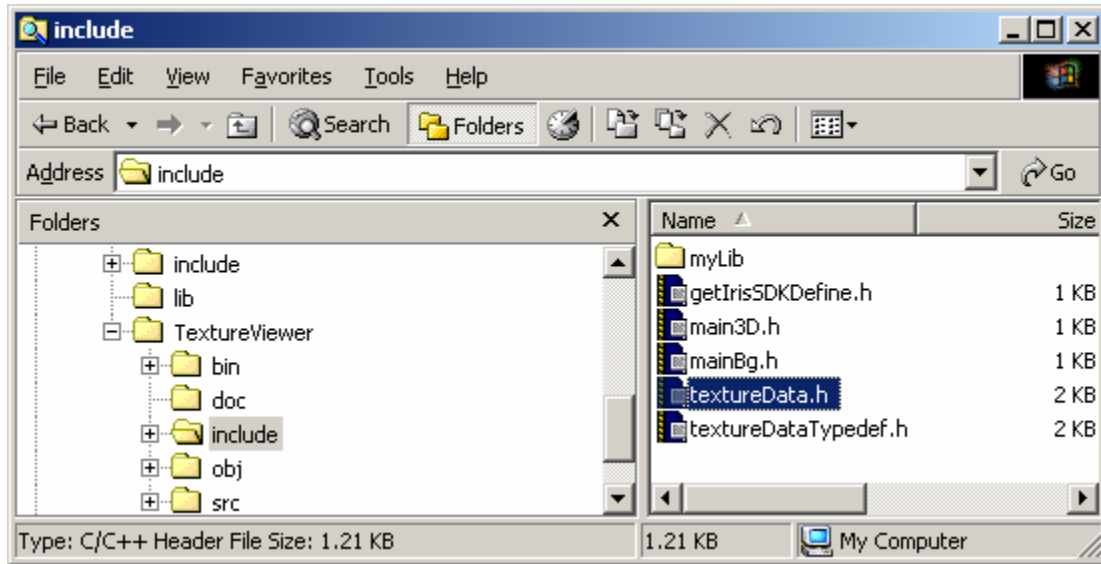


Figure 2-4 Copying textureData.h to TextureViewer

(5) **Execute make.**

Start Cygwin and compile TextureViewer. Input `make` in the TextureViewer directory and press the Enter key to compile.



Figure 2-5 Cygwin Execution Screen

(6) Confirm whether the executable file was created.

Confirm whether an executable file was created by compiling. If compile was successful, `main.srl` and other files will be created in the following location. If these files do not exist, the compile failed.

The executable files are stored in the `\TextureViewer\bin\ARM9-TS\Release` folder.

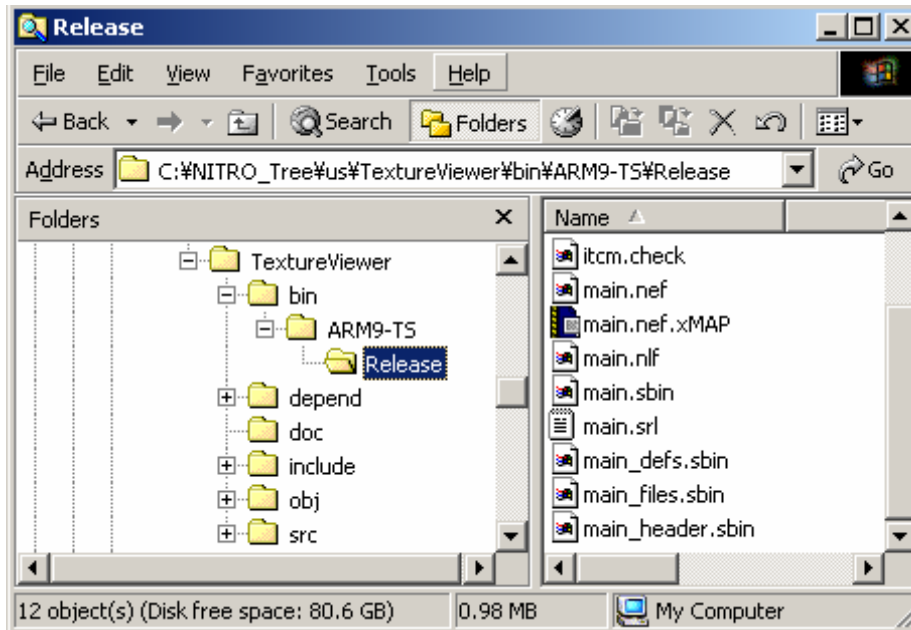


Figure 2-6 Verifying main.srl

(7) Verify operation on an emulator.

Confirm operation of the created executable file with either IS-NITRO-EMULATOR or ensata. If there are no problems, the stored texture will be displayed. See the respective emulator manual to load the file into the emulator.

3 TextureViewer Operation

TextureViewer operation on the NITRO is as follows.

Table 3-1 TextureViewer Operation

Button Operation	Operation
START	Texture information show/hide toggle
L + START	Reset camera position
R + START	Texture DECAL/MODULATE toggle
SELECT	Operation Menu show/hide toggle
R + SELECT	Palette color 0 Color Value/Transparent toggle
A	Show next texture
B	Show previous texture
L + A	Zoom (Move along z-axis of camera (+))
L + B	Zoom (Move along z-axis of camera (-))
L + UP arrow	Tracking (Move along camera/target y-axis (+))
L + DOWN arrow	Tracking (Move along camera/target y-axis (-))
L + LEFT arrow	Tracking (Move along camera/target x-axis (-))
L + RIGHT arrow	Tracking (Move along camera/target x-axis (+))

4 Precautions for Use

TextureViewer was created with the NITRO-SDK. Consequently, the versions of TextureViewer and the NITRO-SDK must coincide. To verify the supported NITRO-SDK version, see the `TextureViewer\ReadMe.txt` file.

© 2004 NINTENDO

The duplication, copying, transferring,
distribution, or loaning of all or part of the
contents of this document is forbidden without
receiving a license from Nintendo Co. Ltd.