NITRO Software Development Environment Setup Guide

(ensata version)

Version 1.3.0 8/3/2004

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

NTR-06-0026-001-A3
Released: September 2, 2004

Table of Contents

NTR-06-0026-001-A3
Released: September 2, 2004

Revision History

Version	Revision Date	Description
1.3.0	8/3/2004	Revisions reflecting version changes for ensata: • ELF files (.nef) are no longer readable • Revisions reflecting NITRO-SDK version upgrade. • NITRO ROM image file extension changed from ”.bin” to ”.srl”.
1.2.0	6/30/2004	• Revisions reflecting NITRO Devkit version upgrade • Version changes for each file • Revisions reflecting changes in ensata interface
1.1.0	4/12/2004	• Changed from IRIS-SDK to NITRO-SDK. • Changed each file version. • Changed the order of sections 3.1 and 3.2.
1.0.0	1/20/2004	Initial release.

1 Preparation for Setup

1.1 Requirements for Setup

The following item is required in order to set up the software development environment:

- PC (Windows 2000 SP4, DirectX 8 or later installed)

2 Setting Up the Software Execution Environment

2.1 Setting Up ensata

This section explains the NITRO software emulator (ensata) and its setup procedure.

From the www.warioworld.com website, download the latest version of ensata and extract it to any location

The file `ensata.exe`, located in the `ensata\Release` folder, is an emulation application for NITRO software made with the NITRO-SDK.

For details, see `ensata\Readme.txt`..

ensata cannot emulate the functions of NITRO completely. Refer to Help in the right-click menu of the application for implemented or planned functions and operating methods.

2.2 Activating ensata

ensata needs to be activated before use, using the following procedure:

- The first time `ensata.exe` is started, the Activation dialog box is displayed.

  Note: After setup, this dialog box is not displayed again when `ensata.exe` is started. The Activation dialog box is also not displayed when the installation is overwritten by a new version because the previously activated content is not overwritten.

- Connect to https://www.warioworld.com/nitro/nitro_key_gen_ensata_rc_1.asp . Copy and paste the Challenge Key of the Activation dialog box in the Challenge Key field and press the send button, as shown in Figure 2-1.

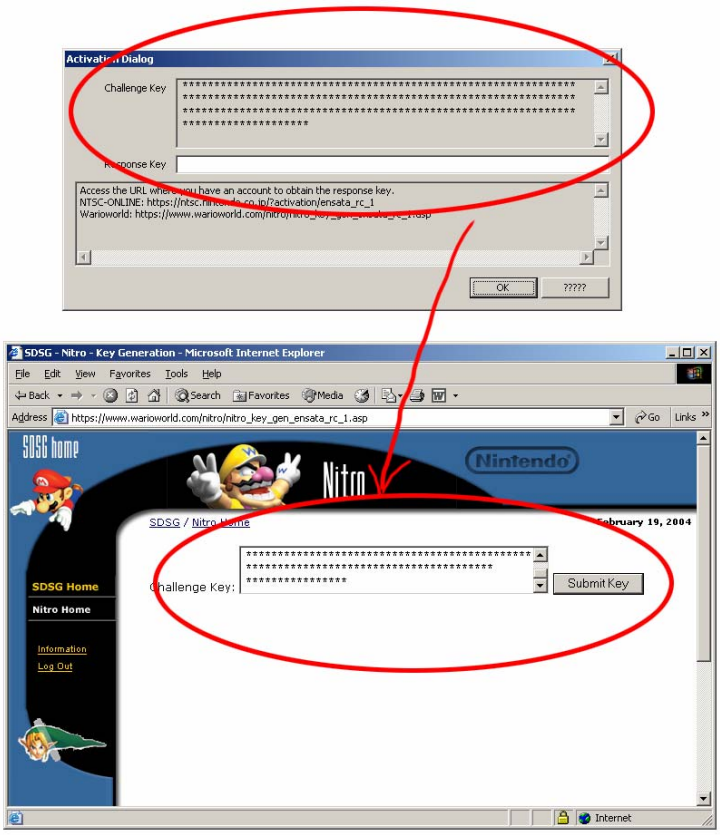

Figure 2-1 Entering the Challenge Key

- Since the Response Key is displayed in the browser, copy and paste a Response Key into the Response Key field in the Activation Dialog and click OK, as shown in Figure 2-2.

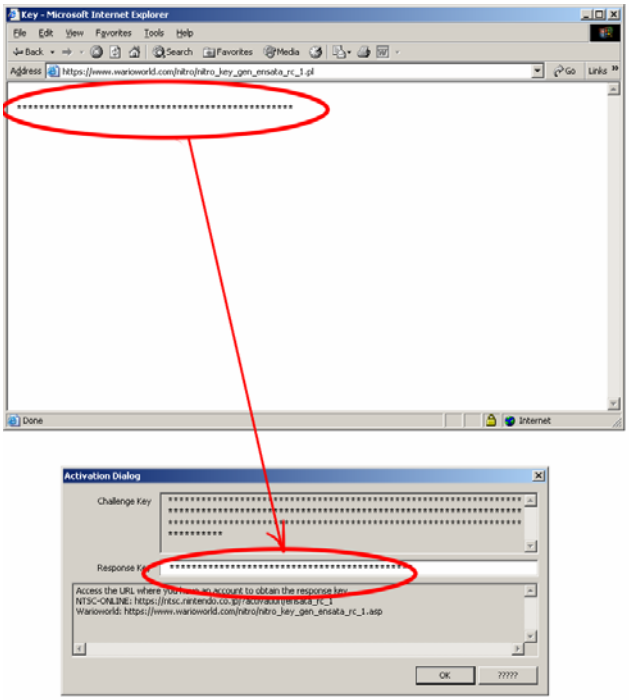

Figure 2-2 Entry of the Response Key

- The control window is displayed. (Figure 2-3)

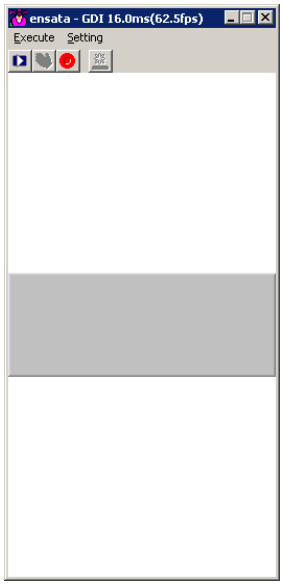

Figure 2-3 The ensata Control Window

This completes the setup for the software execution environment.

3 Setting up the Software Build Environment for ensata Operation

3.1 Setting up the NINTENDO NITRO-SDK

- Download the latest version of NITRO-SDK from the www.warioworld.com website. The `NitroSDK` folder will be expanded.
- Set the path for the expanded `NitroSDK` in the environment variable `NITROSDK_ROOT`.

  Example of setting the environment variable:

  - From the Control Panel, select Environment Variable from the Advanced tab in System Properties.
  - Select User Variables and New.
  - Enter `NITROSDK_ROOT` for the variable name.
  - Enter `C:\NitroSDK` for the variable value.

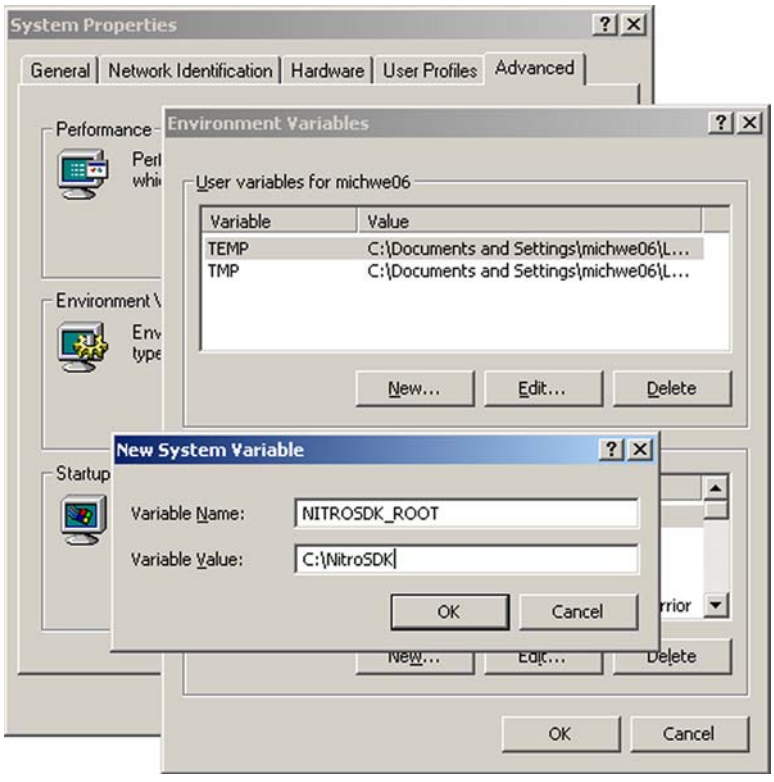

Figure 3-1 Setup of the Environment Variable

This completes the setup of Nintendo NITRO-SDK (hereinafter SDK).

See files under the `NitroSDK\docs` folder for details of the SDK.

3.2 Installing CodeWarrior for NITRO

- Download CodeWarrior for NITRO from the www.warioworld.com website and extract the files.
- Install according to `Developer_Notes - NITRO_Tools.txt`, which is in the unzipped folder.

  Note: CodeWarrior can be installed to any location, but we recommend installing it in a separate folder from other CodeWarrior products so that it does not overwrite previous versions or other CodeWarrior products. If a previous version has not been uninstalled, note the settings of the environment variables for `CWFolder_NITRO`.

This completes the installation of CodeWarrior for NITRO.

In some cases, a dialog box prompting the user to specify a license file is displayed when CodeWarrior for NITRO is used for the first time. In such a case, specify the license file `license.dat` in the folder where CodeWarrior for NITRO has been installed, as shown in Figure 3-3.

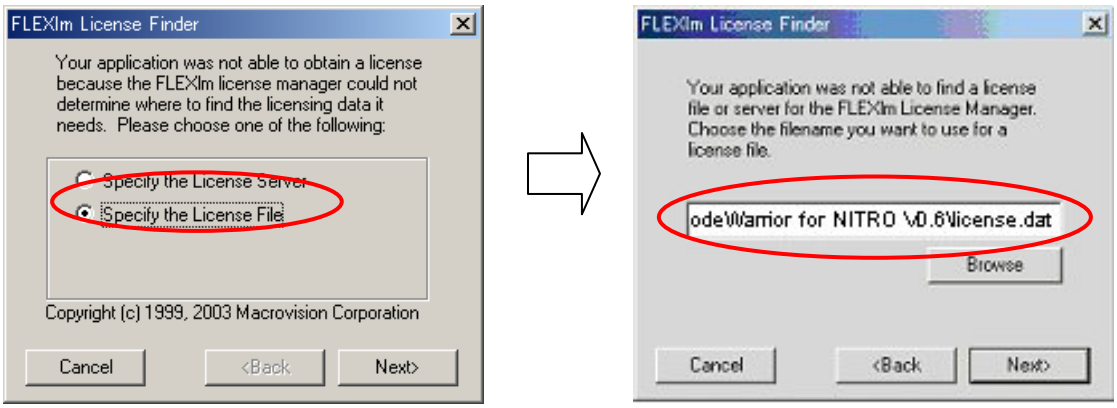

Figure 3-2 FLEXlm License Finder

3.3 Installing the Cygwin Package

The Cygwin package by Red Hat is required for building the NITRO-SDK (execute the makefile).

Obtain and install the Cygwin package as described in `CygwinPackageList.rtf` in the `NitroSDK\docs\SDKTools` folder.

If CodeWarrior's IDE is used exclusively, Cygwin is not required.

This completes the setup of the software build environment.

4 Software Build/Emulation/Print Debug

4.1 Building the Software

NITRO has two CPU cores: ARM9 (ARM946E-S) and ARM7 (ARM7TDMI). It is necessary to create a program for each CPU core, combine the two programs with the ROM header, and create a ROM image file.

The user creates a program for the ARM9 side and combines it with the program already created by the SDK (an ARM7 program), to create the NITRO ROM image file

ensata emulates ARM9 operation. ensata can execute the NITRO ROM image file (`.srl`). There is no need to create a program for the ARM7. (If a program for the ARM7 is created, the ARM7 portion of the combined image file does not work.)

See the document files in `NitroSDK\docs` for details of programming NITRO programs.

Note: Be sure to use the SDK when creating a program for NITRO.

4.2 Using Print Debug

Do the following when creating a program if print debug is used with ensata:

- Include the library header file `nitro.h` in the program.
- Use the `OS_Printf()` function where you want to print.
- Link and build the library `libos.a`.

Note: This is the same print debug method that is used with a combination of the TS board, IS-CGB-EMULATOR hardware, and IS-NITRO-EMULATOR software.

A simple example is shown in `NitroSDK\docs\QuickStartForSDK.rtf`.

4.3 Emulation Procedure

The following is the emulation process for ensata:

- Start `ensata.exe`.
- From the “Execute” menu, select Open NITRO File, and open the NITRO ROM image file (`.srl`), as shown in Figure 4-1.

Figure 4-1 Open the NITRO ROM Image File of ensata

- Click the “Start execution” button, as shown in Figure 4-2.

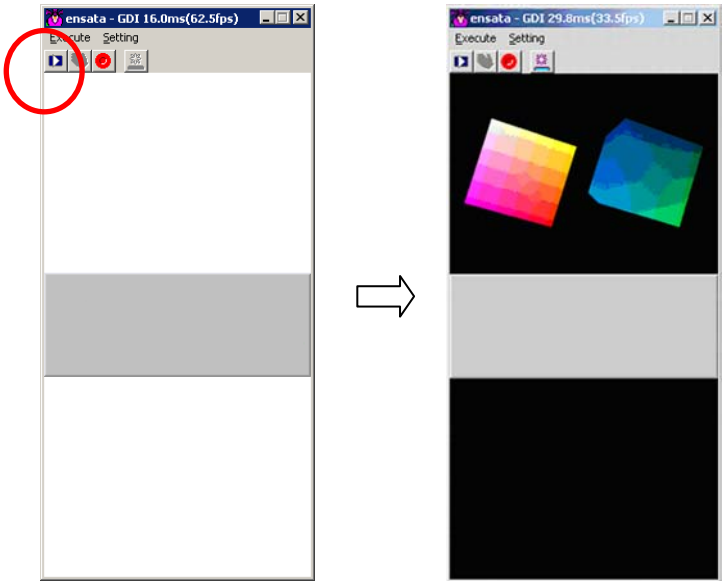

Figure 4-2 Run of ensata

See ensata Help for details and information on using other functions.

This completes the emulation procedure for an ARM9 program.

4.4 Print Debug Procedure

The following is the print debug procedure with ensata:

- Create a program according to the procedure shown in 4.2 Using Print Debug.
- Start `ensata.exe`.
- From the “Execute” menu, select “Read NITRO file” and open the NITRO ROM image file (`.srl`).
- From the “Settings” menu, select “Debug output” and open the debug output window, as shown in Figure 4-3.

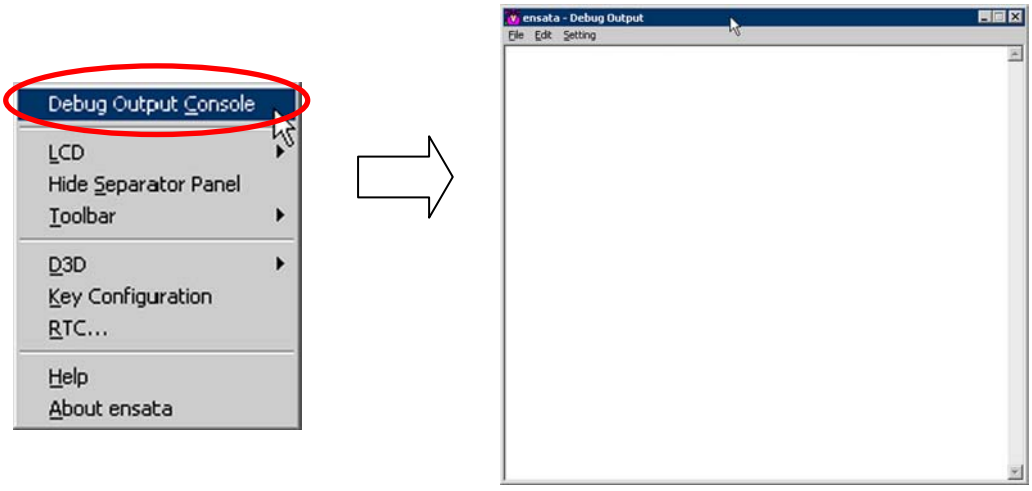

Figure 4-3 Debug Output Window Display

- Click on the “Start execution” button (Figure 4-2).

By performing the above steps, the program is emulated on the ARM9 side and, at the same time, the print debug output used in the program is displayed in the debug output window. (Figure 4-4)

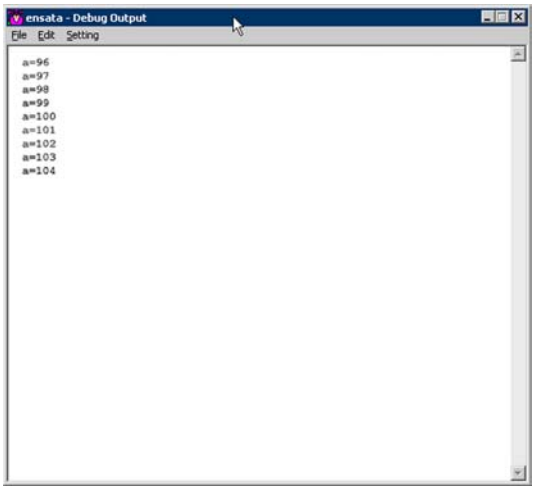

Figure 4-4 Print Debug Output

Refer to ensata Help for details and how to use other functions.

5 Operation Confirmation

5.1 Running the Sample Program

- Set up everything as indicated in Part 3.
- Run `make` on the command line in the `NitroSDK` folder.

By performing the above steps, NITRO ROM image files for all the sample programs in the SDK will be created.

- If the emulation of `NitroSDK\build\demos\gx\UnitTours\3D_Pol_LightColor\bin\ARM9-TS\Release\main.srl` proceeds as described in 4.3, the display on ensata's control window should look like Fig. 5-1.

Figure 5-1 Sample Operation Screen