NINTENDO

NITRO-System

Overview of Tools and Libraries

Version 0.2.5

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

NTR-06-0093-001-A2
Released: September 7, 2004

Table of Contents

Figures

Tables

Revision History

Version	Revision Date	Description
0.2.5	8/2/2004	• Added SOFTIMAGE|XSI and 3ds max in Figure 2-1.
0.2.4	6/30/2004	• Revised Figure 2.1. • ”Nitro-Viewer”, “Nitro-Viewer controller” adopted as official names. • Flowchart for sound data creation revised. • *Concerning the Release of Source Code* updated to reflect current practice. • Corrected spelling.
0.2.3	5/24/2004	• Revised “Preface”. • Revised Figure 2.1. • Added “Photoshop Plugins” to “Tool Overview”. • Added a trademark.
0.2.2	4/12/2004	• Revised Figure 2.1. • Added description of intermediate file plug-ins. • Deleted a description of 2D Graphics Library data grouping. • Deleted the name SAM Manager in the 2D Graphics Library. • Corrected spelling.
0.2.1	4/8/2004	• Revised trademark expressions.
0.2.0	4/6/2004	• Revised Preface. • Revised sound tool content. • Revised library block diagram. • Revised 2D Graphics Library content. • Revised 3D Graphics Library content. • Revised NITRO-Composer content. • Deleted description of Game Framework. • Made other minor revisions.
0.1.1	3/23/2004	• Standardized 2D tool expressions to “NITRO-CHARACTER”.

1 Preface

NINTENDO NITRO-System is the generic term for the tools and libraries that Nintendo has prepared for developing game software for NITRO. Game software developers should be able to devote themselves to creating game content. So the objective was to develop this system to provide game software developers with basic tools and libraries that can be used for much of the game software they develop. This document provides an overview of the tools and libraries that are provided in NITRO-System, to enable you to grasp an overall image of NITRO-System.

2 Tools and Libraries

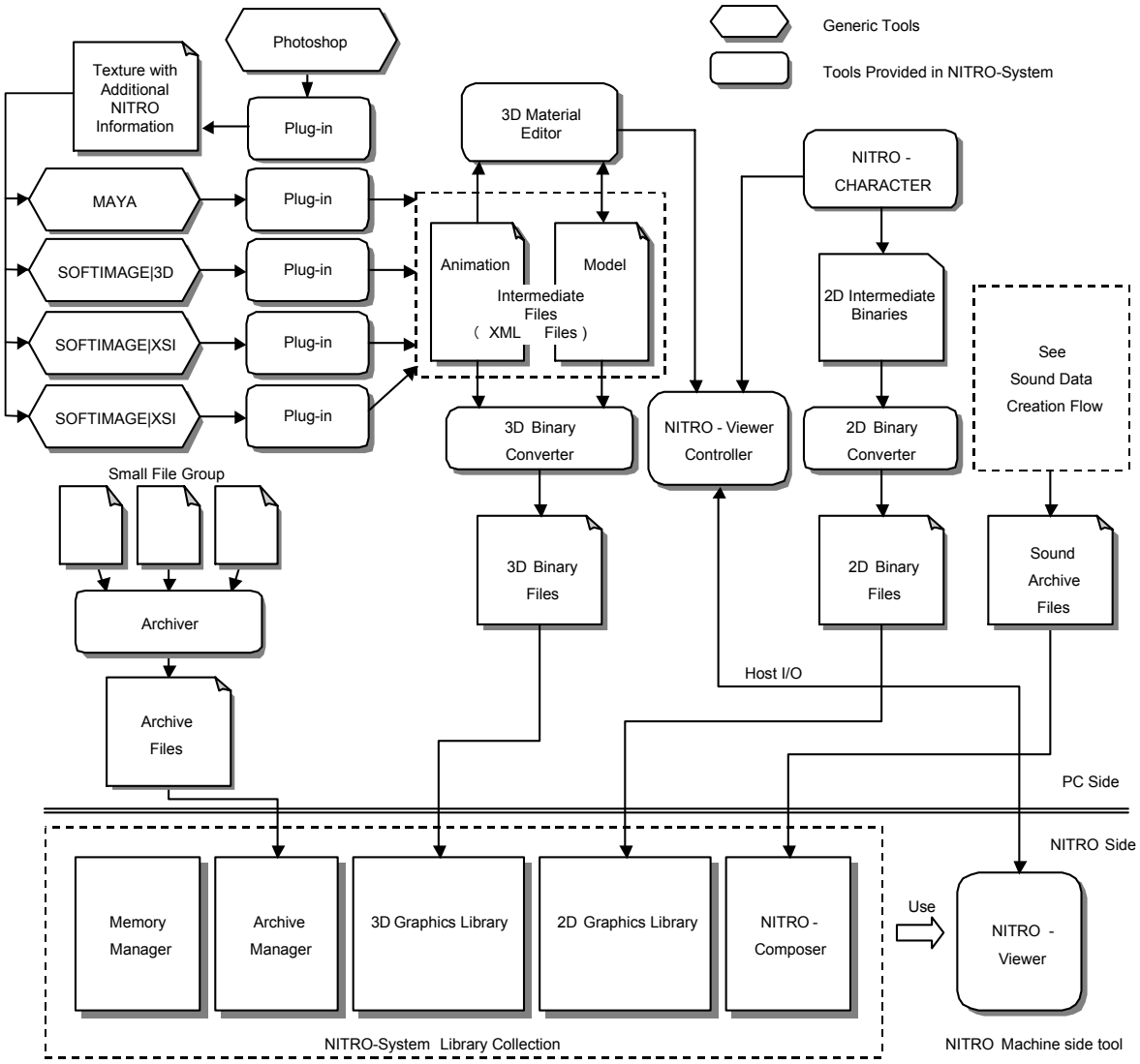

Figure 2-1 Tool and Library Structure

Figure 2-1 shows a simple view of how the tools and libraries presented in NITRO-System cooperate. Figure 2-2, on the following page, shows the flow for making sound data.

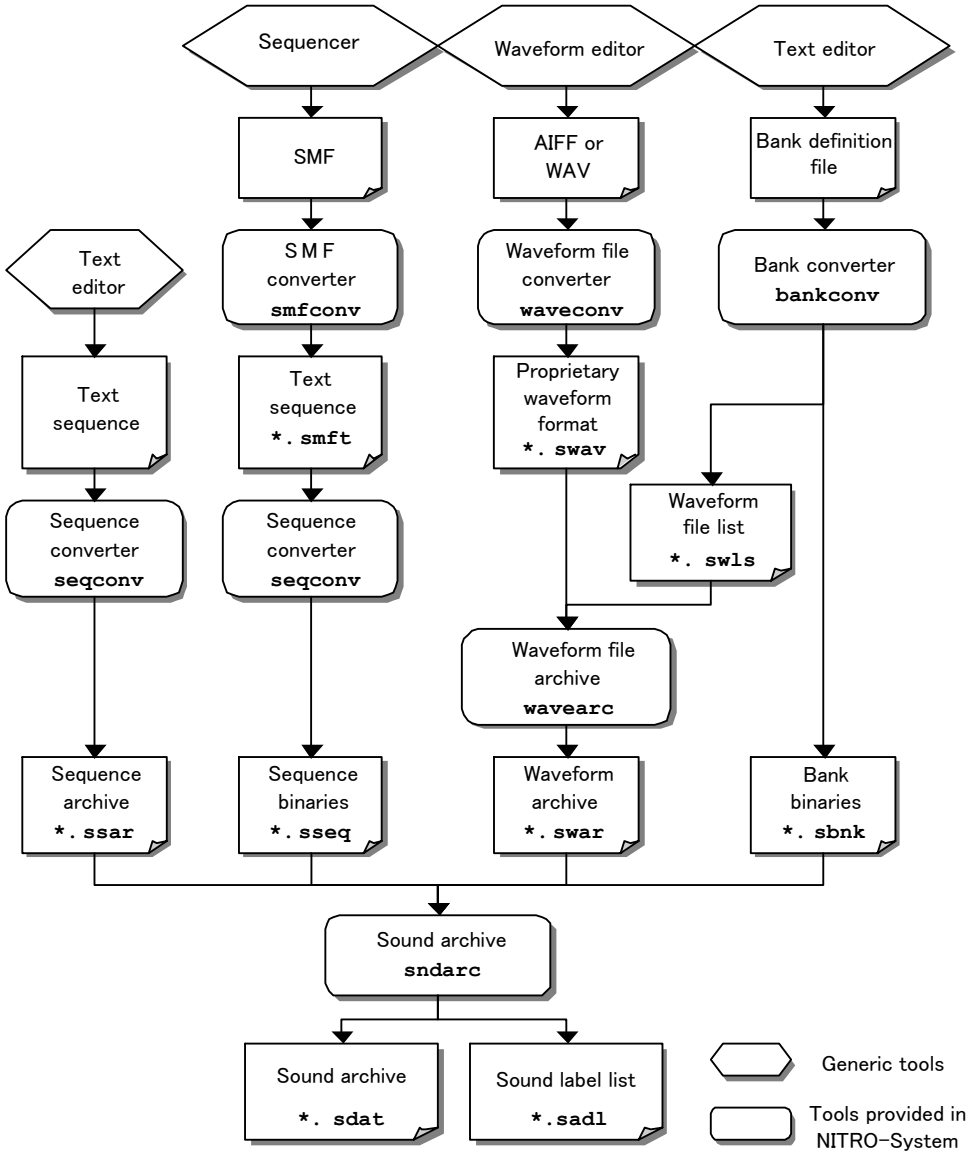

Figure 2-2 Flow for Making Sound Data

3 Tool Overview

This section describes the tools that are provided in NITRO-System.

3.1 NITRO Intermediate File Plug-ins

NITRO-System defines intermediate files (XML-format text files) that store 3D model data and its animation data in a data structure that is optimized for NITRO. You can use a NITRO intermediate plug-in to output an intermediate file from each type of 3DCG tool. NITRO-System provides intermediate file plug-ins for Maya, SOFTIMAGE|3D, SOFTIMAGE| XSI, and 3ds max. (SOFTIMAGE|3D is not supported for the US market. Please contact support@noa.com if you wish to develop using SOFTIMAGE|3D.)

3.2 Photoshop Plug-ins

These plug-ins are used for making texture files for 3D CG tools with Windows versions of Adobe Photoshop versions 6.01, 7.01, and CS (8.01). There are two types of plug-ins. The Filter plug-ins show Photoshop image quality as it will appear when displayed on the retail product. The File Format plug-ins save and load texture files that contain additional NINTENDO NITRO-System information.

3.3 3D Material Editor

The 3D Material Editor is a Windows application. The 3D Material Editor uses a proprietary plug-in to load intermediate files that were output from a 3DCG tool. It can confirm model structure, edit material settings, and so forth. It writes back the edited material settings to the intermediate file.

You can use the 3D Material Editor as a standalone application. Using the NITRO Viewer, you can seamlessly perform a preview as it will appear when displayed on the retail product. When a 3D draw command is issued, a binary converter is called internally. The binary converter converts an XML format intermediate file into a 3D binary file.

3.4 3D Binary Converter

The 3D Binary Converter is a conversion program that converts XML format intermediate files into binary format files that the 3D Graphics Library can use.

- One converter can convert all types of 3D intermediate files.
- It is possible to package multiple model data into one binary file.
- It is possible to load a converted binary file and output a CSV summary.

3.5 NITRO-CHARACTER

NITRO-CHARACTER is a Windows application that can create the character data, screen data, and OBJ data that constitute a NITRO 2D screen. NITRO-CHARACTER has the following functionality:

- Creates character data (can draw and edit dot pictures).
- Creates screen data.
- Creates OBJ animation.

3.6 2D Binary Converter

The 2D Binary Converter is a conversion program that converts the 2D data files that were output by NITRO-CHARACTER into a binary data format that can be processed by the 2D Graphics Library.

3.7 NITRO-Viewer

NITRO-Viewer is an application that runs on an NITRO retail product. It displays the image on the NITRO retail product according to commands and data that are sent from the NITRO-Viewer controller via the Host I/O. NITRO-Viewer accepts 2D and 3D draw commands, and can simultaneously display 2D and 3D images on the screen.

3.8 NITRO-Viewer Controller

The NITRO-Viewer Controller is an application that runs as a window on the Windows side, in order to send draw commands to NITRO-Viewer. The NITRO-Viewer Controller unifies draw commands sent to it from each graphics tool, and uses the Host I/O to transfer this to the NITRO-Viewer that is running on the NITRO retail product. In this way, multiple graphics tools may use the NITRO Viewer, simultaneously.

3.9 Archiver

The NITRO development environment contains a ROM file system. If the file system were to store small files individually, as separate files in ROM, there would be many files, and management would become difficult. Therefore, NITRO-System provides a generic Archiver that makes it possible to collect small files into a large file called an archive file, and register them in the file system.

3.10 Sound Tool (NITRO-Composer)

NITRO-System provides a sound data creation environment. The following tools are provided for creating sound data.

Table 3-1 Sound Tool Collection

Type	Tool Name	Overview
Sound Player	`SoundPlayer`	With this tool you can confirm the converted sound data on the NITRO machine. It is provided with a simple menu screen, so you will be able to select and play sequences.
Sound Archiver	`sndarc`	This tool converts all sound data and collects it in one file. Normally the user will be able to create sound data, using only this tool.
SMF Converter	`smfconv`	Converts SMF files (format 0 or 1) into proprietary text format sequence files.
Sequence Converter	`seqconv`	Converts proprietary text format sequence files into binary files.
Bank Converter	`bankconv`	Converts a text format bank definition file that defines a sound source data set into a binary.
Wave File Converter	`waveconv`	Converts AIFF or WAV files into a proprietary waveform format. When converting you can select any of these formats: 16bitPCM, 8bitPCM, or ADPCM.
Wave Data Archiver	`wavearc`	Collects waveform files converted by the waveform file converter into one file.

4 Library Overview

This section describes overall NITRO library construction, and an overview of the NITRO-System Library.

4.1 Library Structure

The NITRO-System Library is divided into the Foundation Library and High-Level Libraries (High-Level Library Collection). This block diagram shows NITRO-System Library construction.

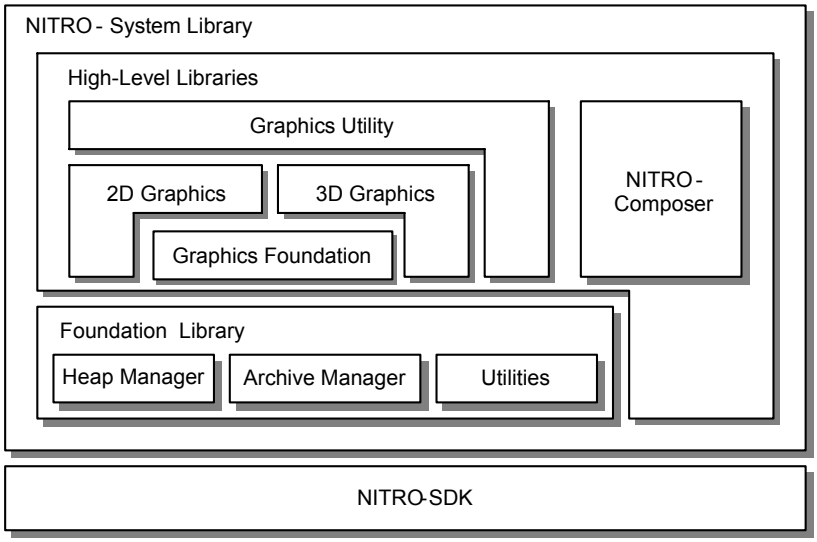

Figure 4-1 NITRO-System Library Block Diagram

4.2 Foundation Library

The Foundation Library provides the following system-related managers and functions:

- Heap Manager
- Archive Manager
- Utilities

4.2.1 Heap Manager

In order to effectively use a 4-megabyte memory, it would be useful if there were some memory management mechanism. We are providing three Heap Managers with special game software features.

4.2.1.1 Extended Heap Manager

The Extended Heap Manager allows you to freely allocate and release memory, just like the `malloc()` function and the `free()` and other functions in the standard C library. In addition to the basic memory allocation and releasing functionality, it adds a number of features for use in game software. These include alignment specification, and grouping and collectively releasing allocated memory blocks.

- It can allocate memory blocks from the front and back of heap memory.
- It can specify the alignment of the memory blocks you want to allocate.
- It is possible to change the size of allocated memory blocks.
- It is possible to attach a group ID to each memory block.
- It is possible to collectively release all allocated memory blocks.
- It is possible for a user to process allocated memory blocks using a callback function.

4.2.1.2 Frame Heap Manager

The Frame Heap Manager is a very simple memory manager. It can only allocate memory blocks that have a specified size, and can only simultaneously release all allocated memory blocks. However, because blocks do not contain any management information, it is memory-efficient.

- Does not require a memory management region.
- Allocates memory blocks from the front and back of the heap memory with no spaces between.
- Cannot individually release allocated memory blocks.
- Can collectively release all allocated memory blocks.
- Can specify alignment for the memory blocks you want to allocate.
- Can store memory block allocation status and return to it.
- Can reduce the total size of a heap to match the amount of memory currently used.

4.2.1.3 Unit Heap Manager

The Unit Heap Manager is a very simple memory manager. It can only allocate memory blocks of the size that is specified when the unit heap is created. That is, this memory manager is for allocating and releasing fixed-size memory blocks. Because there is no management region in the memory blocks, memory usage is efficient.

- Can only allocate fixed-length memory blocks.
- Does not require a memory management region.

4.2.2 Archive Manager

We have installed a ROM file system in the NITRO development environment. If the file system were to store small files individually, as separate files in ROM, there would be many files, and management would become difficult. Therefore NITRO-System provides a mechanism that makes it possible to collect small files into a large file, called an archive file, and then store it in ROM. With the Archive Manager, you can access the data you need that is inside an archive file that has been loaded into main memory.

- You can use an index to access the data in the archive.
- You can use hierarchical directory information and access data in the archive by path name.
- You can reduce the size of the archive file by not including directory information.

4.2.3 Utilities

In addition to the above managers, the Foundation Library provides a bi-directional linked list that can be used by (for example) both game software and multiple libraries.

4.3 High-Level Libraries

This library collection provides functionality that will be useful in facilitating NITRO game software development. Since the libraries are mutually independent, you only need to use the library that has the functionality you want.

4.3.1 Graphics Foundation Library

This library manages NITRO graphics hardware, so that the 2D Graphics Library and the 3D Graphics Library will work well together.

4.3.1.1 VRAM Manager

The VRAM manager is a simple memory manager that manages the NITRO VRAM memory region. This VRAM manager only manages the utilization status of the VRAM region (information as to whether or not it is in use).

- Manages the VRAM region divided into fixed-length memory blocks.
- Manages memory block utilization status with bit fields.

4.3.1.2 VRAM Animation

This function supports animation that is performed by rewriting the content of texture images, color palettes, etc., that are located in VRAM.

4.3.2 2D Graphics Library

The 2D Graphics Library was prepared to simplify representing 2D on NITRO. We created the 2D Graphics Library so that you can easily use 3D graphics functionality to create great 2D games, even if you are not familiar with the NITRO 3D graphics engine.

4.3.2.1 Software Sprites

Software Sprites uses the 3D graphics engine to draw sprites. When you use the highly flexible 3D graphics engine, you can overcome the numerous restrictions that exist in OBJ functionality in the 2D graphics engine. Software Sprites has the following features.

- It can display a maximum 1536 sprites.
- It is possible to rotate, enlarge, and reduce all software sprites.
- With software sprites it is possible to enlarge size to more than double.
- With software sprites it is also possible to rotate around the X axis and the Y axis.
- With software sprites it is possible to set translucency levels 0-31.

4.3.2.2 You Can Use OBJ and Software Sprites Equivalently (Renderer)

You can use the same API to draw with software sprites and with 2D graphics engine OBJ functions. You can also draw the same character with software sprites or with OBJ. You can even dynamically change between drawing using software sprites and OBJ. (You must have character data for software sprites and OBJ in VRAM.)

4.3.2.3 Support for Both the Main Screen and the Sub Screen

With the 2D Graphics Library you can handle the NITRO main screen and sub screen as one continuous image. You can freely set the positional relationships of the main screen and the sub screen. (You must have the character data for the main screen and the sub screen in VRAM.)

4.3.2.4 Can Play NITRO-CHARACTER Animation Data

Using NITRO-CHARACTER, which is provided in NITRO-System, you can use editable animation data to pattern-animate game characters. You can apply this animation to 2D graphics engine OBJ functions and Software Sprites.

4.3.2.5 Support for Multiple Composed Models

You can assemble small pattern-animation elements that have different numbers of frames to make one large game character. This makes it possible to create a large game character that uses less memory than pattern-animating a large character. Also, because you can pattern-animate each element with a different timing, you can create a game character that has extremely complex movement.

4.3.2.6 OAM Emulate Feature

This manages multiple software sprites, and uses the software sprite functionality to emulate the 2D graphics engine OAM. Using this OAM emulate function, you can handle software sprites the same way that you handle OAM.

4.3.3 3D Graphics Library

We made the 3D Graphics Library so that all you need to do to play on a NITRO machine, is use a converter to convert NITRO-System intermediate format file model data and animation data. When you use this library, it will be easy to draw 3D model data that has been output from the material editor, which is provided in NITRO-System, on a screen.

4.3.3.1 Distinctive Features of the 3D Graphic Library

- Supports joint/material/VRAM animation.
- Supports sharing textures between multiple models.
- Models that have the same skeletal structure but differ in size can use common animation data.
- Joint, material, texture, color palette name referencing is possible (maximum 16 characters).

4.3.3.2 Supported Animation Functions

The 3D Graphics Library supports playing the following animation, which is defined in this NITRO-System intermediate format.

- Joint animation functions.
- Material color animation functions.
- Texture SRT animation functions.
- Visibility (show / hide) animation functions.
- Texture pattern animation functions.

4.3.4 Graphics Utility Library

We are currently planning a Graphics Utility Library that will use NITRO graphics functionality and will provide functionality that will be useful for developing game software. (Details to be determined.)

4.3.5 NITRO-Composer

NITRO-System provides the following library collection for developing sound. While it provides the same level of convenience as AGB MusicPlayer2000, it is also component-based. Therefore, customization is comparatively simple. Here is an overview of NITRO-Composer:

- Can use AIFF or WAV waveform files.
- Can use ADPCM. Therefore, it is possible to reduce the size to 1/4 with the same level of sound quality as 16-bit PCM.
- Like AGB, it can play PSG square waves and noise.
- Can use standard MIDI files for sequence files.
- Can record sequences as text.
- No difference in handling sequence data for BGM or effects sounds.
- Maximum number simultaneous sound outputs is 16 (14 when using the capture function).
- Can play up to 16 simultaneous sequences.

The following is a simple introduction to the components that constitute the NITRO-Composer library.

4.3.5.1 ARM7 Side Library

ARM7 processes the sequencer and the synthesizer. Users will not directly handle this library. This library is built into the NITRO-SDK.

4.3.5.2 ARM7/ARM9 Interface Library

This library controls the ARM7 Side Library from the ARM9 side. It can play and stop a sequence, change volume and tempo, etc. Normally users will not directly handle this library. This library is built into the NITRO-SDK.

4.3.5.3 Sound Player Library

This library plays and manages sequence sound. Normally users will use this library to play and stop a sequence.

4.3.5.4 Sound Archive Library

This library extracts specified data from a sound archive, which is a lump of sound data.

5 Concerning the Release of Source Code

This section describes Nintendo's policy on releasing source code for NITRO-System.

5.1 Purpose of Releasing Source Code

Nintendo releases NITRO-System source code mainly to help in confirming library functionality and debugging application programs. Although you may modify library source code for your use, Nintendo cannot guarantee the behavior of modified libraries. Furthermore, in principle, it cannot support such code.

5.1.1 Source Code That Will Be Released

In general, we will release NITRO-System Library source code. However, items that are difficult to release, such as those related to security, may not be released.