NintendoWare for CTR

Sound Development Environment Overview

2010/07/29

Version 1.2

PROVISIONAL TRANSLATION

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

Table of Contents

Figures

1 Introduction

This document describes the sound development environment provided by NintendoWare for CTR (NintendoWare).

Note: This is based on a NintendoWare for Revolution document. Some descriptions do not match the implementation. The Viewer and SoundPlayer features have not been implemented.

1.1 Introduction to NintendoWare Sound Development Environment

The NintendoWare sound development environment is used to carry out sound development for CTR game software. It consists of the runtime library (SoundLib), NW4C SoundMaker (SoundMaker), and command-line tools for conversion of various sound data types.

Note: All library and tool names used in this document are temporary. These names may be changed in the future.

1.2 Features of the NintendoWare Sound Development Environment

The NintendoWare sound development environment has been designed by combining elements of the development environments used for the Nintendo GameCube™, Nintendo DS, and Wii provided in the past.

This section describes the main features of the NintendoWare sound development environment.

1.2.1 Sound Archives

Resource data used by SoundLib is combined into a single collection of data called a *sound archive*.

Since gathering all the data (wave files) used in software streaming would result in a very large archive, such data is managed at the wave file level, instead of including it in the sound archive.

1.2.2 GUI Tools (SoundMaker)

The NintendoWare sound development environment provides SoundMaker, a dedicated tool to support the entire process of creating sound data.

Compatible with Viewer and MCS, SoundMaker is used to check sounds on a CTR system connected to a development PC.

You can also check some sounds during their creation, using emulation on PC.

1.2.3 Development Environment Usable with Individual Sounds

SoundLib, included in the NintendoWare sound development environment, uses AX (temporary name)—the standard audio engine of the CTR-SDK—as its base. Although some operations of SoundLib depend on the NintendoWare system library, with some simple revisions, it can be used to carry out game development for CTR using software other than NintendoWare.

1.2.4 Data Compatibility

As with most conventional sound development, WAV files and AIFF files are the standard formats used for wave files during data creation in the NintendoWare sound development environment. Since standard MIDI files are supported for sequence playback data, these resources can use conventional data as-is.

In addition to these file formats, the NintendoWare sound development environment provides features for effectively utilizing data created in previously supplied development environments by providing some import and file-format compatibility.

As an example of the effective use of data created in a previous development environment, code listings called sequence commands (used when coding sequences using text in the NintendoWare sound development environment) are compatible with NITRO-Composer sequence commands created in a Nintendo DS development environment. Sequence command code listings created for the Nintendo DS using NITRO-Composer can therefore be used as-is in this environment.

1.2.5 Changes from NintendoWare for Revolution

The following changes have been made since the NintendoWare for Revolution sound tools.

1.2.5.1 (Converter) Support for Parallel Conversion

You can now convert sounds in parallel if they do not depend on each other. The number of sounds that can be converted simultaneously is configurable from the **Options** screen.

1.2.5.2 (Converter) Ability to Suppress String Table Output

Although NW4R SoundMaker forced string tables to be output to sound archives, you can now prevent this by opening the **Project Settings** and clearing the **Output String Tables** checkbox in the **Sound Archive** tab. This allows you to reduce the size of sound archives when sounds (and other items) are not called using strings.

1.2.5.3 (Converter) Specification Change for Sound Archive Map Files

The conversion process now generates HTML map files, rather than text files, for sound archives.

1.2.5.4 (Converter) IDs Given to Wave Sound Sets and Sequence Sound Sets

Item IDs are now given to wave sound sets and sequence sound sets, allowing you to specify these items when you load data.

1.2.5.5 (Converter) Specification Change for Item IDs

Information that shows an item's category is now embedded in the upper 8 bits of each ID for the various sound types (sequence sounds, wave sounds, and stream sounds), sequence sound sets, wave sound sets, banks, wave archives, groups, and players.

1.2.5.6 (Converter) Specification Change Related to Cache Output

Intermediate binary files are now always output to the cache folder. However, bank include files (`.cinl`) and intermediate text sequence sounds (`.cseq`) generated by SMF are still output to the same location as the source files.

1.2.5.7 (SoundMaker) Specification Change for Preview Playback

The specifications for playing back previews of sounds and instruments have been reviewed. The following changes were made to playback of sound previews.

- The spacebar registers a sound with the preview player and starts playback.
- Double-clicking the list does not cause a sound to be registered and played back with the preview player.

The following changes were made to playback of instrument previews.

- Pressing F9 or F10, or double-clicking the list, does nothing.
- Hold playback only occurs while the spacebar is held down.

1.2.5.8 (SoundMaker) Wave Archive Nodes Added to the Project Tree

Wave archive nodes have been added to the project tree. You can double-click on a wave archive node to display the wave archive tab.

1.2.5.9 (SoundMaker) Wave Archive List Added

A wave archive list has been added to the sound set tab. To display it, double-click a wave archive node in the project tree. The wave archive list shows the following three items: **Name**, **Separate Loading**, and **Comments**.

1.2.5.10 (SoundMaker) Dialog Box Added for Creating a New Wave Archive

A dialog box has been added for creating a new wave archive. You can select **Add Item** or **Insert Item** from the context menu in the wave archive list to display a dialog box that allows you to choose **Name** and **Separate Loading**.

1.2.5.11 (SoundMaker) Output Format Column Added to the Group List

The group list's **Output Format** and **Wave Sharing** columns have been deleted and the **Output Format** column added. You can set the output format to **Link**, **Embed**, or **No Output**.

1.2.5.12 (SoundMaker) Multi-Bank Dialog Box Added to the Sequence List

You can select **Multi-bank Settings** from the sequence list's context menu to display a multi-bank dialog box. You can specify up to four banks.

1.2.5.13 (SoundMaker) Preview Bank Window Added

The Preview Bank window has been added. To display it, select **Preview Bank** from the **View** menu. Input MIDI signals from a MIDI sequencer are played back using the bank registered in this window.

1.2.5.14 (SoundMaker) Specification Change Related to the Target Bank for MIDI Keyboard Input

The active bank panel is now the target bank for MIDI keyboard input. No audio is played if there are no active bank panels. Buttons to display the bank and instruments were removed from the toolbar together with this change.

1.2.5.15 (SoundMaker) Specification Change Related to MIDI Settings

You can now choose from MIDI keyboard input and MIDI sequencer input. The active bank panel's bank is used to play back MIDI signals from the device configured for MIDI keyboard input. The

Preview Bank window's bank is used to play back MIDI signals from the device configured for MIDI sequencer input.

1.2.5.16 (SoundMaker) Import of NW4R SoundMaker Projects

You can import the following files from NW4R SoundMaker projects: project (`.rspj`) files, sound-set (`.rsst`) files, bank (`.rbnk`) files, and text-sequence sound (`.rseq`) files. The original files are not changed when you import them; new files are saved using a different extension in the same folder.

Text-sequence sound files are imported as follows.

- A batch replace is used to change the strings "`.rseq`" and "`.rinl`" to "`.cseq`" and "`.cinl`" within the file.
- The "`.rseq`" files are imported recursively.

1.2.5.17 (Library) Playback of Wave Sounds, Sequence Sounds, and Stream Sounds

You can now play back wave sounds, sequence sounds, and stream sounds with PcSDK. For details, see the `simple` sample demo.

2 Folder Structure

The following figure shows the folder hierarchy of sound-related tools and documents provided by NintendoWare.

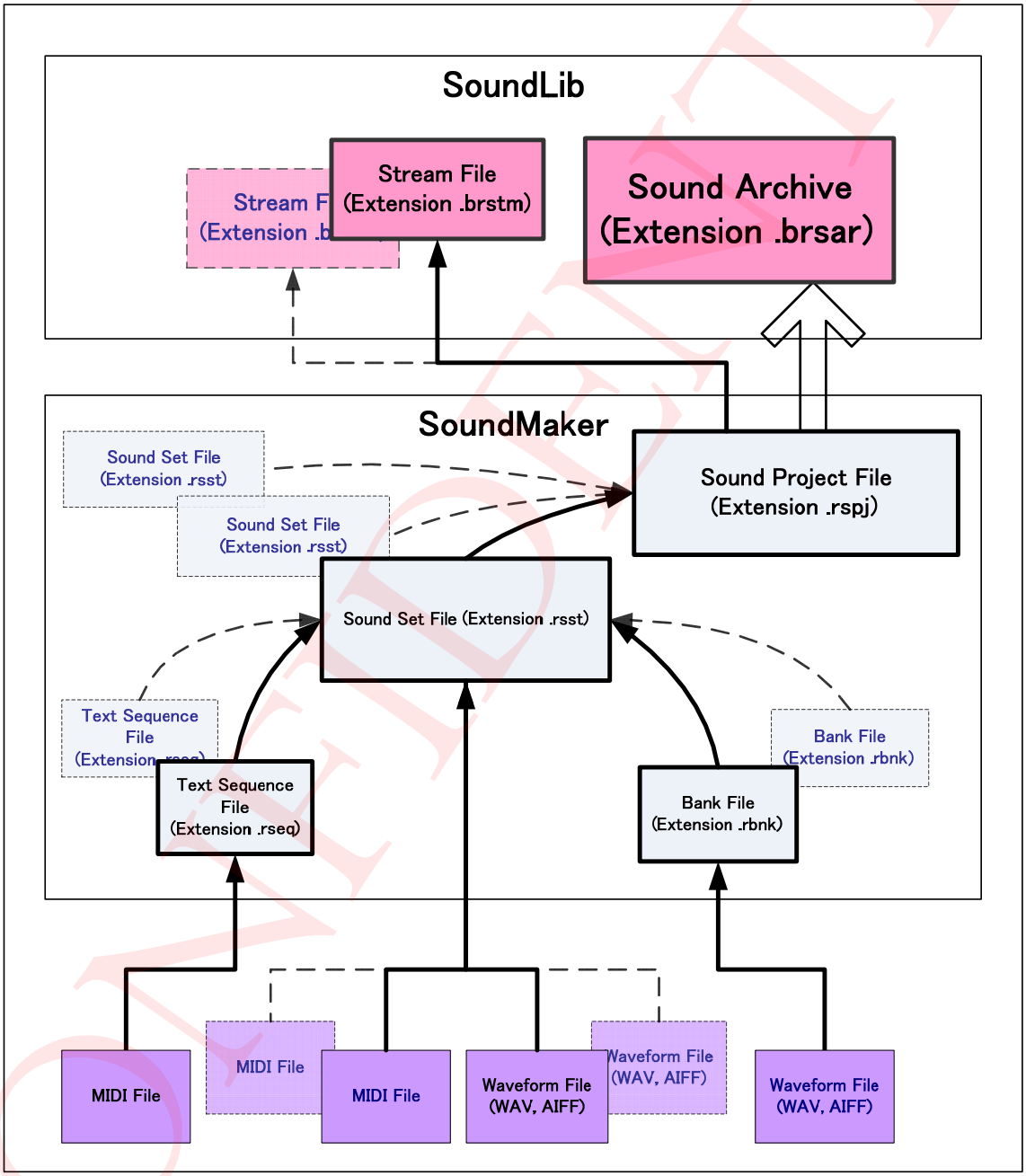
Figure 2-1 NintendoWare Folder Hierarchy

```
NintendoWare
└ CTR （Environment Variable: NW4C_ROOT）
  ├ demos
  │  └ snd                                  // Sound Library Demo Data
  ├ documents                               // Various Documents
  │  ├ Overview
  │  │  └ Sound_OverView.pdf                // Sound Development Environment Overview (this document)
  │  ├ ProgrammersGuide
  │  │  └ Sound_ProgrammersGuide.pdf        // Sound Programmer's Guide
  │  ├ ReleaseNotes                         // Release Notes
  │  ├ Sound
  │  │  ├ Sound_SequenceFileFormat.pdf      // Sound Sequence File Format
  │  │  └ Sound_SystemManual.pdf            // Sound System Manual
  │  └ SoundMaker
  │     ├ SoundProjectConverterManual.pdf   // Sound Project Converter Manual
  │     └ SoundMaker.html                   // SoundMaker User's Guide
  ├ libraries                               // Libraries
  └ tools                                   // NW4C Tools
     └ SoundMaker
        ├ NW4C_SoundMaker.exe               // SoundMaker
        ├ NW4C_SequenceConverter.exe        // Sequence Converter
        ├ NW4C_SmfConverter.exe             // SMF Converter
        ├ NW4C_SoundProjectConverter.exe    // Sound Project Converter
        └ samples                           // SoundMaker Sample Data
           └ simple
```

3 Sound Development Environment Configuration

3.1 Flow of Sound Development

Figure 3-1 Flow of Sound Development



3.1.1 Creating Data

MIDI sequence data and wave data are created using commercially available sequencers and wave-editing software, respectively. SoundMaker uses a runtime library to create a usable sound archive from the standard MIDI files (SMF files) and wave files (AIFF and WAV files) output by such software.

3.1.2 SoundLib Features

SoundLib uses sound archives to play and control each sound and to perform memory management.

SoundLib can be used to play three types of sound files: stream sounds, sequence sounds, and wave sounds.

Sound control utilizes a mechanism known as a *sound handle* to control sounds having different replay formats transparently. At the same time, each sound is managed by a player. The number of voices to be played and the priority of voices are controlled at the player level.

SoundLib implements an original method of memory management called a *sound heap*. Sound data is loaded into memory in groups that have been set by the sound designer.

4 Main Sound Representations Under NintendoWare

4.1 Supported Replay Formats

4.1.1 Stream Sounds

Stream sounds are played back while loading wave data into memory. It is also possible to play back more than one stream sound at once. Playback is possible using only the minimum amount of memory required, even if the wave data is long, because the entire data does not have to be located in memory.

Stream sounds are suitable for playing extended wave data, such as BGM or other sounds.

4.1.2 Sequence Sounds

Sequences can be played back using sequence data and sound source data. Data converted from a standard MIDI file can be used as sequence data. Properties such as tempo and volume by track can be freely changed when using sequence playback. You can also keep the data size required relatively small.

Sequence sounds are suitable for playback of BGM.

4.1.3 Wave Sounds

Although wave data is played in order to replay wave sounds as with stream sounds, wave data must be loaded into memory ahead of time. This reduces the load, compared to stream sounds, when performing repetitive playback because there is no need to load data each time.

Wave sounds are suitable for the playback of relatively small amounts of wave data, such as sound effects.

Note: Currently, wave sounds are supported only when playing back simple wave files based on the specification of a wave file and initial parameters.

Future plans for wave sounds call for the addition of performance features such as superimposition of more than one wave data, the ability to continually change each parameter along the time axis, and the ability to simultaneously vary multiple parameters such as volume and tone based on input of a single parameter.

4.2 3D Sounds

3D sound can create spatial representations such as distance and angle by using sound emissions based on the direction and coordinates of the *3D sound listener* (a microphone in the real world), coordinates of the *3D sound actor* (a sound source in the real world), and the sound request made for the 3D sound actor.

3D sound is ideal for representing sounds in games with characters that move around freely in 3D space.

5 Sound Runtime Library

5.1 snd Library

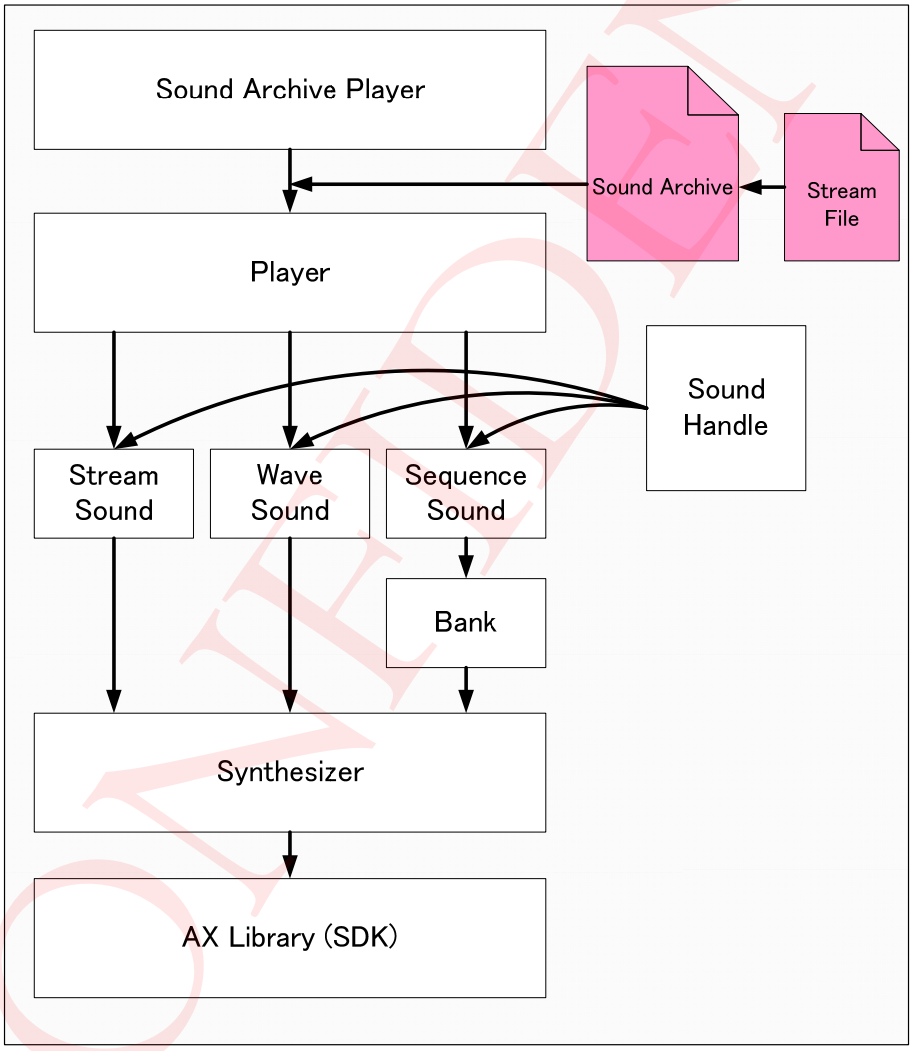
The snd library is designed to replay sound data created for the sound runtime library on the CTR system. The snd library is included in SoundLib.

The snd library is a C++ class library. Each sound feature is implemented by class.

5.2 snd Library Configuration

The following diagram illustrates the configuration of the snd library.

Figure 5-1 snd Library Configuration Diagram



5.2.1 Sound Archive Player

The sound archive player class is central to the snd library. Programmers normally play sounds using a sound archive player function.

A sound archive player can play back a sound archive (a collection of sound data) using a player to replay the sounds.

5.2.2 Sound Archives

A sound archive class corresponds to the data in a single sound archive. Sound archives are used to handle data input and data analysis of sound archive data located on DVD.

5.2.3 Sound Handles

Various forms of playback control using handles are available for sounds played on a player, including pausing and changing the audio volume.

Since the same interface is used for sound handles, whether the sound being controlled is a sequence sound or stream sound, you do not need to be aware of differences in the types of sounds.

Another handle, called a *sequence sound handle*, is also provided to control sequence sounds and support operations, such as changing tempo, and can only be employed when using sequence sounds.

5.3 Memory Management of Sounds

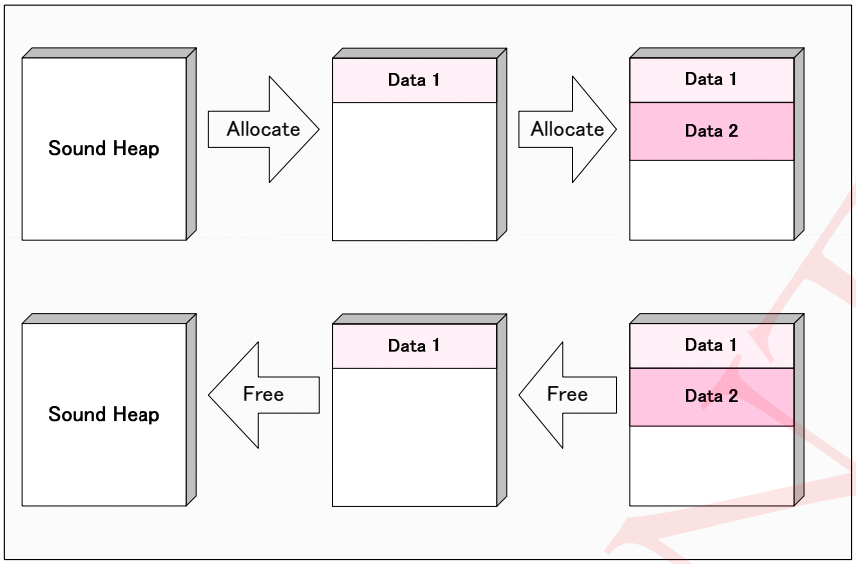
Memory management of sound data is provided using a sound heap, which is an easy way to use a heap system that sound designers can use.

5.3.1 Sound Heap

A sound heap is used to store sound data in a sound archive.

Using a heap system, memory is allocated in order from the top of the sound heap. When deleting unneeded data, it is freed in the order of most recently allocated.

Figure 5-2 Allocating and Freeing the Sound Heap

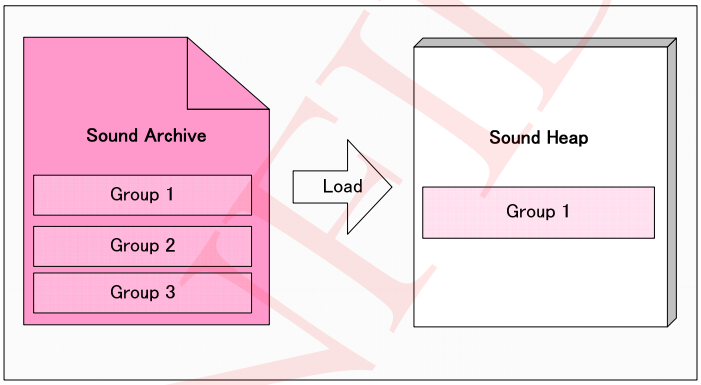


5.3.2 Groups

Sound data stored in sound archives is separated into groups.

Sound designers separate multiple sets of sound data into groups each time data is loaded. Programmers load sound data into the sound heap at the group level.

Figure 5-3 Loading Groups



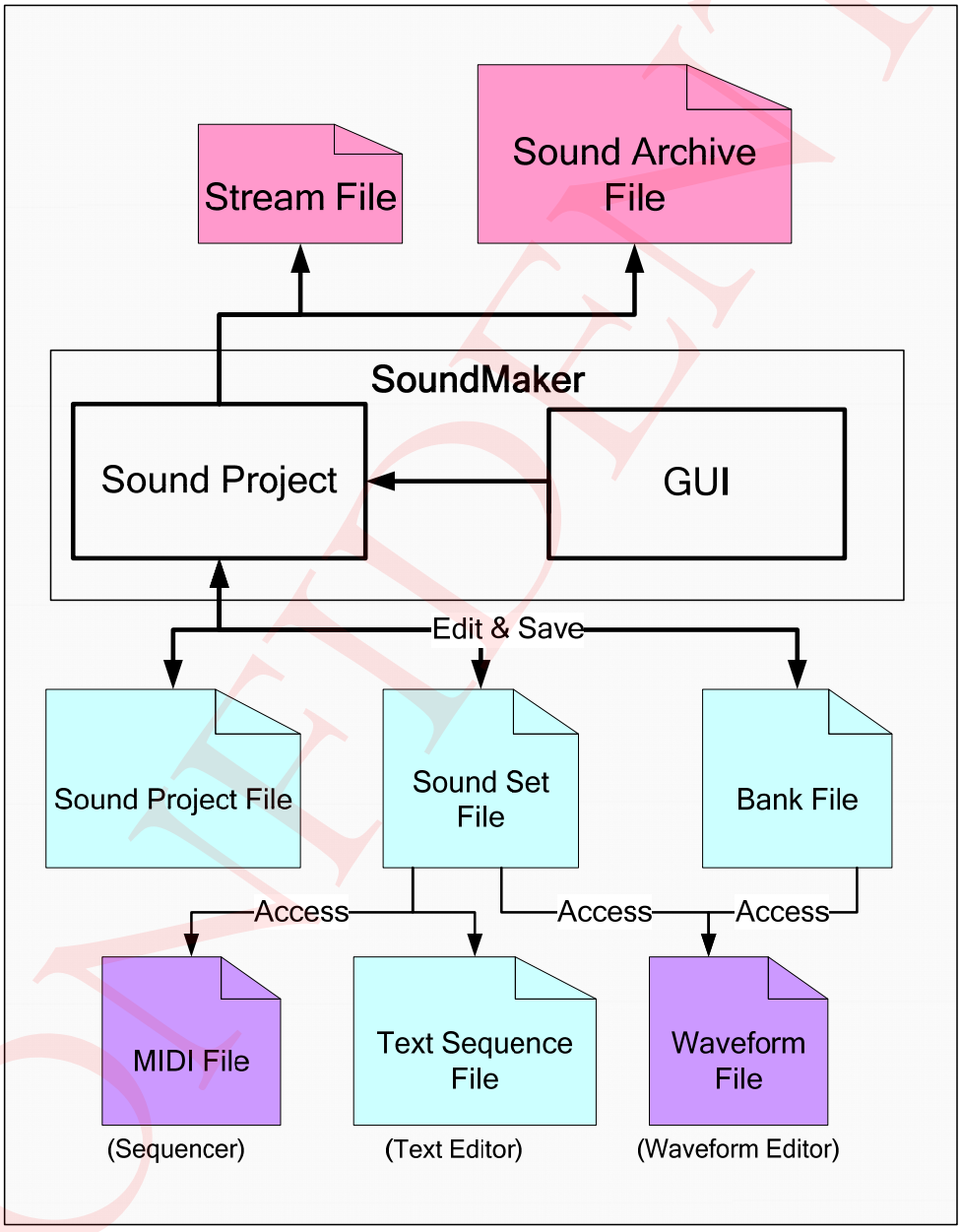
6 GUI Tools

6.1 Sound Data Development Using SoundMaker

SoundMaker is a special NintendoWare GUI tool that runs on Windows.

The following conceptual diagram illustrates the flow of data when using SoundMaker.

Figure 6-1 SoundMaker Operational Concept 1



6.1.1 Linking with Existing Applications

There are many excellent existing applications for sound development available, with each sound designer using his or her favorite, depending on the application. SoundMaker emphasizes linking with these frequently-used, existing applications.

The wave file formats supported are WAV files (`riff` files) most commonly used on Windows and AIFF files commonly used on Macintosh. These wave files are used in the creation of stream sounds, sequence sound banks, and wave sounds.

Standard MIDI files, supported by nearly all sequencers, are supported for playback data.

SoundMaker is intended to not only use these resources, but also provide full sound creation support through the addition of features specialized for game development.

6.1.2 Creating Sound Archives

Sound projects are project files created using SoundMaker. One sound project corresponds to one sound archive.

Sound projects can define small-scale collections of sounds called *sound sets* as multiple references Unique settings, such as the heap size required for the sound archive, are made for the sound project, and the creation of each actual sound is carried out using sound sets.

The sound set accesses the files required for each sound: MIDI files, wave files, bank files used as sound sources by MIDI files, and the text sequence file in which sequence commands have been coded.

The converter creates the sound archive, using the sound project and all files referenced by it.

6.1.3 Linking using Multiple Staff

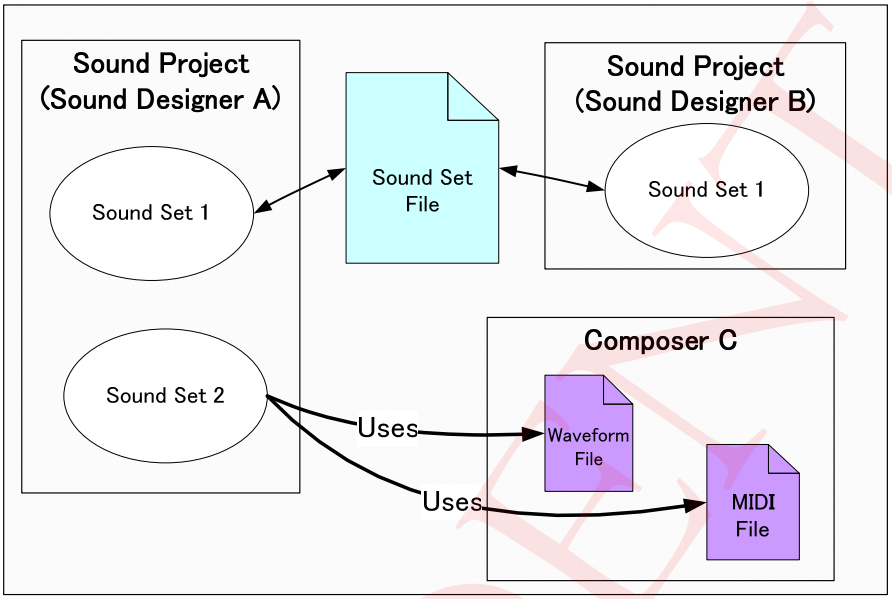
All sounds are saved in the sound sets (sound set files) accessed by a sound project, including the following:

- Stream sounds
- Wave sounds
- Sequence sounds
- Information regarding the specified player to be used by each sounds

Nearly all this information can be used to build the sound archive, including which banks are accessed by sequences.

In addition to the ability to share data and pass it back and forth using wave files and MIDI files, data can be shared more efficiently when a staff is working cooperatively using SoundMaker, dividing responsibilities and work at the sound set file level.

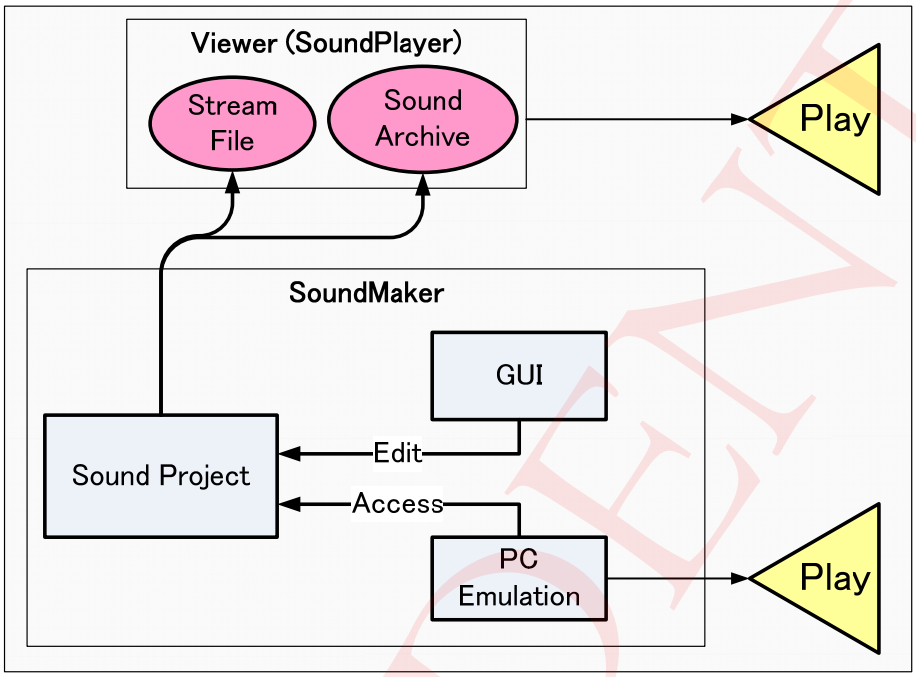
Figure 6-2 Example of Collaboration Using SoundMaker



6.1.4 Checking Each Sound

SoundMaker provides two methods of checking sounds during creation: playback on a CTR system and playback using PC emulation.

Figure 6-3 SoundMaker Operational Concept 2



6.1.4.1 Playback on the CTR System

Viewer (or SoundPlayer) is launched when converted sound archives are passed to Viewer (or SoundPlayer), which then allows sounds in the archive to be played on the CTR system and controlled with the system controller.

6.1.4.2 Playback Using PC Emulation

SoundMaker can be used to check sounds while they are being created using PC emulation.

PC emulation supports MIDI and can be used to play instruments from MIDI keyboards and sequencers connected to the PC.

7 Revision History

Version	Revision Date	Description
1.2	2010/07/29	• Changed the document format
1.1	2009/11/11	• Added Changes from NintendoWare for Revolution. • Updated the description of the folder structure. • Deleted unnecessary Revolution-centric text.
1.0	2009/10/30	• Initial version.