

N I N T E N D O
NITRO-System
NITRO-Composer
Release Notes

Version 1.12.0

**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	About NITRO-Composer	12
1.1	Multi-thread Operation	12
2	Major Changes	13
2.1	Changes from the 2006/05/29 Version	13
2.1.1	Changes to the Sound Data	13
2.1.1.1	Added a Feature to SoundPlayer for Changing the Output Effects	13
2.1.1.2	Added a Feature to SoundPlayer for Displaying the Processing Load of Streams	13
2.1.1.3	Added a Feature to SoundPlayer for Simultaneous Playback of Multiple Sounds	13
2.1.1.4	Added a Feature to Specify the Alignment of Sound Archives	13
2.1.1.5	Added a Feature to Specify the File Types to Convert within a Sound Archive	13
2.1.2	Changes to the Program	13
2.1.2.1	Added Sound Archive Stream-related Functions	13
2.1.3	Bug Fixes	13
2.1.3.1	Fixed a Bug that Caused a Conversion Error due to Command Line Argument Overruns	13
2.2	Changes from the 2005/09/01 Version	14
2.2.1	Changes Related to Sound Data	14
2.2.1.1	Added the Mute Sequence Command	14
2.2.1.2	Back-reference Labels with Sound Archive Definitions File	14
2.2.1.3	Expansion of Label Names Embedded in SMF Text Commands	14
2.2.2	Changes Related to the Program	14
2.2.2.1	Added the Function NNS_SndPlayerSetTrackMuteEx	14
2.2.2.2	Added the Function NNS_SndCaptureDestroyThread	14
2.2.3	Bug Fixes	14
2.2.3.1	Losing Final Wait Command when Converting SMF Format 1	14
2.2.3.2	Paths with Space Characters were not Converted Properly	14
2.2.3.3	Breakdown of SoundMap File Layout when Group Number Created with Missing Number	15
2.3	Changes from the 2005/06/06 Version	15
2.3.1	Changes Related to Sound Data	15
2.3.1.1	Playing Back Stream Data in Memory	15
2.3.1.2	Added Option for Specifying an Output Filename for the Sound Archiver sndarc	15
2.3.1.3	Added Option for Specifying Preprocess Files for the Sound Archiver sndarc	15
2.3.2	Changes Regarding the Program	15
2.3.2.1	Added NNS SndArclnitWithResult Function	15
2.3.2.2	Added Functions for Getting the Capture Status	15
2.3.3	Bug Fixes	16
2.3.3.1	Fixed Frequent Failure of NNS_SndUpdateDriverInfo	16
2.3.3.2	Fixed File Path Misinterpretation with sndarc and bankconv	16
2.4	Changes from the 2005/03/28 Version	16
2.4.1	Changes Related to Sound Data	16

2.4.1.1	Added Instrument Setting to Ignore Note Off	16
2.4.1.2	Original Key Value Setting in Bank Definition File	16
2.4.2	Program Changes	16
2.4.2.1	Deleted Sound Driver (SND) Function Reference	16
2.4.2.2	Added Function to Create Thread for Stream Preparation	16
2.4.3	Bug Fixes	17
2.4.3.1	Fixed Problem with Minor Noise Generation	17
2.4.3.2	Fixed Bug with Portamento when Real-time MIDI was Played on the SoundPlayer	17
2.4.3.3	Bankconv Drive Letter Case-sensitivity Problem	17
2.5	Changes from the 2005/01/31 Version	17
2.5.1	Changes to Sound Data	17
2.5.1.1	Addition of a Feature to Individually Load Waveform Data to the Player Heap	17
2.5.1.2	Addition of a Feature to Use the Player Heap when Playing a Sequence Archive	17
2.5.1.3	Change of the Sound Map File (*.smap) Format	17
2.5.1.4	Expansion of Numeric Value Notation in Text Files	18
2.5.1.5	Addition of the NULL Type Instrument	18
2.5.2	Changes to the Program	18
2.5.2.1	Addition of a Function to Configure Volume for Each Player	18
2.5.2.2	Addition of a Function to Configure the Track Pan Range	18
2.5.2.3	Addition of a Function to Obtain the Stream Data Time Length	18
2.5.2.4	Addition of Functions to Obtain the Sound Driver Information	19
2.5.2.5	Addition of the SND_COMMAND_IMMEDIATE Flag for SND_FlushCommand Function	19
2.5.2.6	Added SND_INST_NULL to SNDInstType	19
2.5.3	Bug Fixes	19
2.5.3.1	Release of Channels Playing Sounds with a Portamento Command	19
2.5.3.2	Data Load Could Not be Selected in Manual Load Mode on SoundPlayer	19
2.5.3.3	Unnecessary Wait Inserted at the End of the Track with smfconv.exe	19
2.5.3.4	Invalid Wait Time Output in smfconv.exe	19
2.5.3.5	False Error Occurrence in bankconv.exe	19
2.6	Changes from the 2004/12/06 Version	20
2.6.1	Changes to Sound Data	20
2.6.1.1	Addition of a Feature to Individually Load Waveform Data	20
2.6.1.2	Addition of a Feature that Limits the Channels a Sequence Uses	20
2.6.1.3	Support for the UNC Format Path Notation	20
2.6.1.4	Addition of the #include<filename> Format	20
2.6.1.5	Addition of the Sequence Command printvar	20
2.6.1.6	Elimination of a Distinction between Capital and Small Letters in the Key Notation	20
2.6.2	Changes to SoundPlayer	21
2.6.2.1	Improvement of the Channel Meter	21
2.6.3	Changes to the Program	21
2.6.3.1	Addition of the NNS_SndCaptureSetPostOutputEffectCallback Function	21
2.6.3.2	Addition of a Feature that Limits the Channels Used by a Sequence	21

2.6.3.3	Addition of the micThrough Demo.....	21
2.6.4	Bug Fixes	21
2.6.4.1	Fixed an SMF Converter Bug.....	21
2.6.4.2	Problem with Allocated Channels being Deallocated by NNS_SndArcStrmStopAll.....	21
2.7	Changes from the 2004/11/10 Version	22
2.7.1	Changes to Sound Data	22
2.7.1.1	Using Two Channels for Mono Stream Playback	22
2.7.2	Changes to the Program	22
2.7.2.1	Added the NNS_SndArcStrmStartEx2 Function.....	22
2.7.2.2	Made the Sound Driver Thread-safe	22
2.7.2.3	Calling NNS_SndArcStrmAllocChannel to the Stream Player Playing Back	22
2.7.3	Changes to Manuals	22
2.7.3.1	Added Caution Regarding the Output Effects	22
2.7.4	Bug Fixes	22
2.7.4.1	Bug that Prevented the Sound Generation of the Note After tieon.....	22
2.8	Changes from 2004/10/12 version	23
2.8.1	Changes to the Program	23
2.8.1.1	Resource Control Functions Added	23
2.8.2	Bug Fixes	23
2.8.2.1	Sample of Stream Playback without Loop.....	23
2.8.2.2	Current Sound Archive Switching Bug During Stream Playback.....	23
2.8.2.3	NNS_SndArcStrmSetChannelVolume Function Bug.....	23
2.8.2.4	Sequence Converter seqconv Command Analysis Bug.....	23
2.9	Changes from the 2004/09/16 Version	24
2.9.1	Changes to SoundPlayer	24
2.9.1.1	IS-AGB-MIDI Support	24
2.9.2	Changes to the Program	24
2.9.2.1	Added Output Effect Feature	24
2.9.2.2	Automated the Sleep Process.....	24
2.9.2.3	Added the Stream Player Specification Function	24
2.9.2.4	Added the Mono Flag Setting Function.....	24
2.9.3	Changes to Sound Data	25
2.9.3.1	Player Label Specification.....	25
2.9.4	Changes to Manuals	25
2.9.4.1	Added Cautions for Use when Changing to the Sleep Mode	25
2.9.4.2	Added Explanation for Not Interrupting Stream.....	25
2.9.4.3	Added Introduction About Output Effects	25
2.9.5	Bug Fixes	25
2.9.5.1	Infinite Loop Bug During Sequence Fadeout	25
2.9.5.2	Mistake in Interpretation of the @PATH Command.....	25
2.9.5.3	Release Channel Process During Sequence Fadeout.....	26
2.9.5.4	Distinction Between Upper and Lower Case in File Names.....	26

2.9.5.5	Waveconv Loop Start Location Check.....	26
2.10	Changes from the 2004/09/02 Version	26
2.10.1	Changes to Sound Data	26
2.10.1.1	Extension of Loop_start Sequence Command.....	26
2.10.1.2	Specifying Loop for Individual SMF Tracks	26
2.10.1.3	SMF Loop Specification	26
2.10.1.4	Extension of .sddl Files and Name Change	26
2.10.1.5	Naming of .sddl Files Unified	26
2.10.2	Changes to the Program	27
2.10.2.1	NNS_SndCaptureCreateThread Function Added	27
2.10.2.2	NNS_SndCaptureStartSampling Function Added	27
2.10.2.3	NNS_SndArcGetSeqArcSeqCount Function Added.....	27
2.10.3	Bug Fixes	27
2.10.3.1	NNS_SndPlayerMoveVolume Function.....	27
2.10.3.2	Unresponsive Sequences	27
2.10.3.3	Incorrect Error Message in SoundPlayer	27
2.10.3.4	Turning Mute Off when Sound Not Muted	27
2.10.3.5	Bank List File Updating	27
2.10.3.6	Sequence Converter seqconv Error Check.....	27
2.11	Changes from the 2004/08/10 Version	28
2.11.1	Real-time MIDI Playback.....	28
2.11.2	Changes to Sound Data	28
2.11.2.1	Bank List File Output.....	28
2.11.2.2	Embedded SMF Text Commands	28
2.11.2.3	Multiple Definition of the Same Label in Sequence Archives	28
2.11.3	Changes to the Program	28
2.11.3.1	Feature for Awakening from the Sleep Mode	28
2.11.3.2	Clearing the Sound Heap.....	28
2.11.3.3	Added the NNS_SndArcSetup Function	29
2.11.3.4	Added the NNS_SndArcStrmSetupPlayer Function	29
2.11.4	Bug Fixes	29
2.11.4.1	Player Heap Release Bug.....	29
2.11.4.2	Track Output Error of SMF Converter smfconv	29
2.11.4.3	Shifting the Loop Starting Point in SMF Converter smfconv	29
2.11.4.4	Path Interpretation Error of Bank Converter bankconv.....	29
2.11.4.5	#define Multiple Definition Error	29
2.12	Changes from the 2004/07/20 Version	29
2.12.1	Support for Stream Playback.....	29
2.12.1.1	Changes to the Sound Archive Definition Ffile	30
2.12.1.2	Changes to the SoundPlayer	30
2.12.1.3	Sound Archive Stream Library Added.....	30
2.12.2	Changes to the Program	30

2.12.2.1	Stream Library Added	30
2.12.2.2	NNS_SndStopSoundAll Function Added.....	30
2.12.2.3	NNS_SndHeapDestroy Function Added	30
2.12.3	Changes to Sound Data.....	30
2.12.3.1	Changes to SMF Converter <code>smfconv</code> Output Label Names.....	30
2.12.4	Bug Fixes	31
2.12.4.1	Sound Archiver <code>sndarc</code> Abnormal Termination	31
2.12.4.2	Bank Converter <code>bankconv</code> Abnormality Error	31
2.12.4.3	Irregular Order of SMF Converter <code>smfconv</code> Meta Event	31
2.13	Changes from the 2004/06/10 Version	31
2.13.1	Changes to the Binary Data Format.....	31
2.13.2	Changes to SoundPlayer	31
2.13.2.1	Name of Executable File has Changed.....	31
2.13.2.2	Changes in the Default Platform	31
2.13.2.3	Sequence Type Selection Feature	31
2.13.3	Changes to the Program	32
2.13.3.1	Waveform Playback Library Added	32
2.13.3.2	Effects Features Added.....	32
2.13.3.3	Changes in the Arguments for <code>NNS_SndCaptureStartReverb</code>	32
2.13.3.4	Changes in the <code>NNSSndArcBankInfo</code> Structure	32
2.13.3.5	Functions Deleted.....	32
2.13.4	Changes to Sound Data.....	32
2.13.4.1	Ability to Use Multiple Waveform Archives per Bank Added.....	32
2.13.4.2	Option to Delete Sound Archive Symbol Data.....	32
2.13.5	Bug Fixes	33
2.13.5.1	Mistake in the Value for the Maximum Number of Simultaneous Sequence Playback	33
2.14	Changes from the 2004/04/12 Version	33
2.14.1	Migration from Previous Environment	33
2.14.2	File System Support.....	33
2.14.2.1	Sound Archive Files	33
2.14.2.2	Embedding Sound Data	33
2.14.2.3	Initialization of the Sound Library	33
2.14.2.4	Memory Management	33
2.14.2.5	ARM7 Component Changes	33
2.14.3	Changes to SoundPlayer	34
2.14.3.1	Changes to the Convert Command.....	34
2.14.3.2	Label Name Display.....	34
2.14.3.3	Manual Loading Mode Added	34
2.14.4	Changes to Player.....	34
2.14.4.1	Playback of Multiple Sequences	34
2.14.4.2	Number of Players	34
2.14.4.3	Specifying the Number of Sequences that Can Be Played Simultaneously	34

2.14.4.4	Player Priority During Fadeout	34
2.14.5	Changes to Sound Data	35
2.14.5.1	Slight Changes to Release/Decay	35
2.14.5.2	Sound Map File Output	35
2.14.5.3	Specifying Index Numbers	35
2.14.5.4	Outputting the Program Number List.....	36
2.14.5.5	Expression Added	36
2.14.5.6	Sweep_pitch Can Be Used from MIDI.....	36
2.14.5.7	Maximum Value for prg and mod_delay Commands Changed	36
2.14.6	Addition of Extended Sequence Commands.....	36
2.14.6.1	Random Commands	36
2.14.6.2	Variable Commands.....	37
2.14.6.3	Control Commands	37
2.14.7	Changes to the Sound Program.....	37
2.14.7.1	Deletion of the NNS_SndArcPlayerInit Function	37
2.14.7.2	Deletion of the NNS_SndArcPlayerStartSeqWithPlayerNo Function	37
2.14.7.3	Function Name Changes	37
2.14.7.4	Sequence Variable Functions Added	37
2.14.7.5	Obtaining the Number of Ticks.....	37
2.14.7.6	Added Other New Functions	37
2.14.8	Bug Fixes	38
2.14.8.1	Voice Priority	38
2.14.8.2	The NNS_SndPlayerSetTrackPan Function.....	38
2.14.8.3	Release Problem with NNS_SndPlayerStopSeq.....	38
2.15	Changes from the 2004/04/07 Version	38
2.15.1	Changes to Sound Data	38
2.15.1.1	Sequence Label	38
2.15.1.2	Preprocessor Directives	38
2.15.2	Changes to the Programs	38
2.15.2.1	Addition of a Track Mute Function.....	38
2.15.3	Changes to the Manuals	38
2.15.3.1	Sound Tools Manual.....	38
2.15.3.2	Overview	39
2.15.3.3	Fixing Errors.....	39
2.15.4	Bug Fixes	39
2.15.4.1	Function for Setting Up Track Parameters	39
2.16	Changes from the 2004/03/18 Version	39
2.16.1	Installing the NITRO-SDK Patch	39
2.16.2	Changes to SoundPlayer	39
2.16.2.1	Directory Changes	39
2.16.2.2	Executable File.....	39
2.16.2.3	Migration from Previous Environment	39

2.16.3	Changes to the Sound Archive.....	40
2.16.3.1	Sound Label List File	40
2.16.4	Changes to Sequences.....	40
2.16.4.1	Changes to Binary Format	40
2.16.4.2	Addition of Envelope Command.....	40
2.16.4.3	Addition of Loop Command	40
2.16.4.4	porta Command from MIDI.....	40
2.16.5	Changes to Manuals	40
2.16.5.1	Sound System Manual.....	40
2.16.5.2	Reference Manual.....	40
2.16.5.3	Sound Programmer Guide	40
2.16.5.4	Envelope Specifications	40
2.16.5.5	Waveform File Format.....	40
2.16.5.6	Pitch Notation.....	41
2.16.5.7	Voice Priority	41
2.16.5.8	Information about SMF Compatibility with MusicPlayer2000	41
2.16.6	Changes to Programs	41
2.16.6.1	Addition of Functions	41
2.16.6.2	Sample Program.....	41
2.16.7	Bug Fixes	41
2.16.7.1	waveconv Error	41
2.16.7.2	seqconv Error.....	41
2.16.7.3	Checking Multiple Definition of Label.....	41
2.16.7.4	Transpose	41
2.16.7.5	Sequence End During Sound Generation.....	41
2.16.7.6	Player Number	42
2.16.7.7	SMF Conversion with sndarc	42
2.16.7.8	Bank Convert with sndarc.....	42
2.17	Changes from the 2004/03/02 Version	42
2.17.1	Changes to the Binary Data Format.....	42
2.17.2	Changes to the Text Sequence Format.....	42
2.17.2.1	Comments.....	42
2.17.2.2	Local Labels.....	43
2.17.2.3	Changes to Sequence Archive Header Section	43
2.17.3	Changes to the Sequence Commands	44
2.17.3.1	opentrack Command.....	44
2.17.3.2	Note Command.....	44
2.17.3.3	Portamento	44
2.17.3.4	Sweep	44
2.17.4	Bug Fixes	44
2.17.4.1	SMF Converter <code>smfconv</code>	44
2.17.4.2	Sequence Converter <code>seqconv</code>	44

2.17.4.3	Sequence Playback Immediately After Startup.....	44
2.18	Changes from the 2004/02/17 Version	45
2.18.1	Addition of Bank Instrument Type SWAV.....	45
2.18.2	Changes to *.swav Filenames Generated from Bank.....	45
2.18.3	Addition of Modulation Range	45
2.18.4	MIDI Control Code Allocation	45
2.18.5	Changes to the Sweep Command.....	46
2.18.6	Adjustment of the Envelope	46
2.18.7	Playback of Multiple Sequence Archives.....	46
2.18.8	Bug Fix of Runtime Error when Sound Archive Section Empty	46
2.18.9	Bug Fix of Player Priority Setup for the Sequence	46
2.19	Changes from NITRO-Composer Specification Proposal	46
2.19.1	Specification Changes for Priority Setup	46
2.19.2	Elimination of Bank Instrument Type PCM	47
3	Known Issues.....	48

Revision History

Version	Revision Date	Description
1.12.0	2007/03/14	Support for the 03/14/2007 version.
1.11.0	2006/05/29	Support for the 05/29/2006 version.
1.10.0	2005/09/01	Support for the 09/01/2005 version.
1.9.0	2005/06/06	Support for the 06/06/2005 version.
1.8.0	2005/03/28	Support for the 03/28/2005 version.
1.7.0	2005/01/31	Support for the 01/31/2005 version. Changed "NITRO" to "Nintendo DS."
1.6.0	2004/12/06	Support for the 12/06/2004 version.
1.5.0	2004/11/10	Support for the 11/10/2004 version.
1.4.0	2004/10/12	Support for the 10/12/2004 version. Added cautions regarding multi-thread operation.
1.3.0	2004/09/16	Changes for the 09/16/2004 version.
1.2.0	2004/09/02	Changes for the 09/02/2004 version.
1.1.0	2004/08/10	Changes for the 08/10/2004 version.
1.0.0	2004/07/20	Changes for the 07/20/2004 version.
0.5.0	2004/06/10	Changes for the 06/10/2004 version.
0.4.0	2004/04/12	Changes for the 04/12/2004 version.
0.3.0	2004/04/07	Changes for the 04/07/2004 version.
0.2.0	2004/03/18	Changes for the 03/18/2004 version.
0.1.0	2004/03/01	Changes for the 03/02/2004 version.
0.0.1	2004/02/17	Initial version.

1 About NITRO-Composer

NITRO-Composer is a sound development environment that includes the tools and libraries that are necessary for developing sound on Nintendo DS. NITRO-Composer is included in NITRO-System. To use NITRO-Composer, you will need NITRO-SDK and the Nintendo DS development environment in addition to NITRO-System.

Before using NITRO-Composer, read through the Overview (`NITRO_Composer_Overview.pdf`) or the Quick Start Guide (`NITRO_Composer_QuickStartGuide.pdf`).

1.1 Multi-thread Operation

The NITRO-System library is not designed to be fundamentally thread-safe (in a form that supports multi-threading.) Therefore, when the NITRO-Composer library API is called from the interrupt handler or a different thread, it may not work properly.

However, the functions of the sound driver (SND) are thread-safe and they can be called from interrupt handler and different threads.

2 Major Changes

2.1 Changes from the 2006/05/29 Version

2.1.1 Changes to the Sound Data

2.1.1.1 Added a Feature to SoundPlayer for Changing the Output Effects

A feature was added to SoundPlayer that allows output effects to be changed.

2.1.1.2 Added a Feature to SoundPlayer for Displaying the Processing Load of Streams

The processing load of the stream thread during stream playback was made to be displayed at the top of the SoundPlayer screen.

2.1.1.3 Added a Feature to SoundPlayer for Simultaneous Playback of Multiple Sounds

A feature was added to play multiple sounds at the same time in SoundPlayer. For details, refer to the *Sound Designer Guide*.

2.1.1.4 Added a Feature to Specify the Alignment of Sound Archives

The --align option was added to the sndarc sound archiver, which makes it possible to specify the alignment of sound data within an archive. For details, refer to the *Sound Tools Manual*.

2.1.1.5 Added a Feature to Specify the File Types to Convert within a Sound Archive

The --convert option was added to the sndarc sound archiver for specifying the file type of files to convert. For details, refer to the *Sound Tools Manual*.

2.1.2 Changes to the Program

2.1.2.1 Added Sound Archive Stream-related Functions

The NNS_SndArcStrmGetChannelCount function, which gets the number of channels of the stream data being played, and the NNS_SndArcStrmSetVolume function, which changes the volume of a stream, were added.

2.1.3 Bug Fixes

2.1.3.1 Fixed a Bug that Caused a Conversion Error due to Command Line Argument Overruns

There were times when a conversion error would occur due to a command line argument overrun when the number of files such as bank files increased, but this has been corrected.

2.2 Changes from the 2005/09/01 Version

2.2.1 Changes Related to Sound Data

2.2.1.1 Added the Mute Sequence Command

The `mute` command was added to configure the mute setting from the sequence data.

This command only works with NITRO-SDK Versions 3.1 and later. For details, see the *Sequence Data Manual*.

2.2.1.2 Back-reference Labels with Sound Archive Definitions File

Previously, you defined Sound Archive definitions file labels ahead of time. Now, you can also use labels that are defined later.

2.2.1.3 Expansion of Label Names Embedded in SMF Text Commands

For the expansion of label names embedded in SMF text commands, you can use `$$` instead of `$` to expand only the track name and not the file name.

For details, see the *Sequence Data Manual*.

2.2.2 Changes Related to the Program

2.2.2.1 Added the Function `NNS_SndPlayerSetTrackMuteEx`

Added the function `NNS_SndPlayerSetTrackMuteEx`, which can mute the sound without stopping it.

This function only works with NITRO-SDK Versions 3.1 and later.

2.2.2.2 Added the Function `NNS_SndCaptureDestroyThread`

Added the function `NNS_SndCaptureDestroyThread`, which ends the capture thread.

2.2.3 Bug Fixes

2.2.3.1 Losing Final Wait Command when Converting SMF Format 1

When SMF format 1 was converted, the `wait` command in the last part of the track was sometimes left out. This problem is fixed.

2.2.3.2 Paths with Space Characters were not Converted Properly

When the conversion process was run for a path that included space characters, the process would sometimes not work properly. This problem is fixed.

2.2.3.3 Breakdown of SoundMap File Layout when Group Number Created with Missing Number

When group numbers were directly specified using the group definitions in the Sound Archive definitions file (*.sarc) and a missing group number was created, a problem could occur with a breakdown of the layout of the SoundMap file (*.smap) that gets output during conversion. This problem is fixed.

2.3 Changes from the 2005/06/06 Version

2.3.1 Changes Related to Sound Data

2.3.1.1 Playing Back Stream Data in Memory

Although playback of stream data contained in a sound archive initialized by `NNS_SndArcInitOnMemory` is impossible, the method that initializes sound archive using `NNS_SndArcInit` can be used to play back stream data.

2.3.1.2 Added Option for Specifying an Output Filename for the Sound Archiver sndarc

An option has been added that enables you to specify filetype of the output (such as, .sdat, .sdl, and .sddl files) from the sound archiver `sndarc`. For details, see the *Sound Tools Manual*.

2.3.1.3 Added Option for Specifying Preprocess Files for the Sound Archiver sndarc

An option has been added to specify as an argument the preprocess files that are processed before the sound archive definition file for the sound archiver `sndarc`.

For details, see the *Sound Tools Manual*.

2.3.2 Changes Regarding the Program

2.3.2.1 Added NNS_SndArcInitWithResult Function

Added `NNS_SndArcInitWithResult()` to determine if sound archive initialization was successful in loading a file. Use this function if you believe file loading may fail. Using this function in place of `NNS_SndArcInit()` informs you of the success or failure of the sound archiver initialization.

For details, see the *Function Reference*.

2.3.2.2 Added Functions for Getting the Capture Status

Added two functions. `NNS_SndCaptureIsActive` retrieves the execution status of a capture and `NNS_SndCaptureGetCaptureType` retrieves the type of the currently executing capture.

For details, see the *Function Reference*.

2.3.3 Bug Fixes

2.3.3.1 Fixed Frequent Failure of NNS_SndUpdateDriverInfo

Fixed `NNS_SndUpdateDriverInfo`, because calling the function failed one out of two times even if called once per frame.

2.3.3.2 Fixed File Path Misinterpretation with `sndarc` and `bankconv`

Fixed file path misinterpretation in which the sound archiver `sndarc` and bank converter `bankconv` would interpret file paths as relative paths not specified using `@PATH`. This issue led to other issues such as inappropriate conversion errors.

2.4 Changes from the 2005/03/28 Version

2.4.1 Changes Related to Sound Data

2.4.1.1 Added Instrument Setting to Ignore Note Off

Added the ability to define an instrument in the bank definition file that ignores Note Off and plays to the end of the waveform data.

In the envelope's release setting, instead of a release value, enter "DISABLE". This will invalidate the release and the sound will play at the same volume until the end of the waveform data even after the Note Off occurs.

For details, see the *Bank Data Manual*.

2.4.1.2 Original Key Value Setting in Bank Definition File

Added the ability to set the original key in the bank definition file using a value instead of key notation.

For details, see the *Bank Data Manual*.

2.4.2 Program Changes

2.4.2.1 Deleted Sound Driver (SND) Function Reference

Included the sound driver (SND) source code and function reference in NITRO-SDK and deleted the sound driver (SND) function reference from the NITRO-System package.

2.4.2.2 Added Function to Create Thread for Stream Preparation

Added `NNS_SndArcStrmCreatePrepareThread`, which starts a thread that performs stream preparation.

For details, see the *Function Reference*.

2.4.3 Bug Fixes

2.4.3.1 Fixed Problem with Minor Noise Generation

If a sequence was started and the volume was changed or another sequence was played immediately thereafter, minor noise was generated in some cases. This was fixed.

This problem occurred when a series of sound functions was called and execution occurred during different sound frames, causing partially ready sounds to be played for an instant. To avoid this problem, the actual processing of sequence data is delayed until the invocation of the series of sound functions is complete.

2.4.3.2 Fixed Bug with Portamento when Real-time MIDI was Played on the SoundPlayer

When playing a real-time MIDI with the SoundPlayer and using portamento, the pitch change started at the wrong position. This was fixed.

2.4.3.3 Bankconv Drive Letter Case-sensitivity Problem

In the bank converter (bankconv), differences in the case of the drive letter caused the wrong file to be recognized. This was fixed.

2.5 Changes from the 2005/01/31 Version

2.5.1 Changes to Sound Data

2.5.1.1 Addition of a Feature to Individually Load Waveform Data to the Player Heap

Revisions were made so the waveform data individual load feature now works when loading to the player heap.

2.5.1.2 Addition of a Feature to Use the Player Heap when Playing a Sequence Archive

Revisions were made so bank data and waveform data can be loaded to the player heap when playing a sequence archive.

However, the sequence archive cannot load to the player heap. The data must be loaded to the sound heap in advance.

2.5.1.3 Change of the Sound Map File (*.smap) Format

A part of the sound map file (*.smap) format was changed, and the heap size consumed when loading data can be output.

For further details, refer to the *Sound Archive Manual*.

2.5.1.4 Expansion of Numeric Value Notation in Text Files

The sound archive definition file, the bank definition file, and the text sequence file were expanded to allow binary and bit notation as numeric value notation for parameters.

With binary notation, prepend a “0b”, as in 0b001001.

Bit notation is useful for describing a numeric value that carries a meaning that a specific bit is 0 or 1, such as a bit flag, and is written in the format { 1,3,6-8 }. The example indicates that the lower bits 1, 3, and the bits from 6 to 8 have a value of 1, or 0b111001010. The lowest bit will be 0.

For further details, refer to the *Sound Archive Manual*, the *Bank Data Manual*, and the *Sequence Data Manual*. (The contents of each are identical).

2.5.1.5 Addition of the NULL Type Instrument

Made it possible to define a NULL type instrument in the bank definition file.

A NULL type instrument does not play any sound. For example, this can be used to create an instrument that does not play any sound for a given split by combining with a key split.

For further details, refer to the *Bank Data Manual*.

2.5.2 Changes to the Program

2.5.2.1 Addition of a Function to Configure Volume for Each Player

The `NNS_SndPlayerSetPlayerVolume` function was added to allow the volume value to be configured for each player.

For further details, refer to the *Function Reference*.

2.5.2.2 Addition of a Function to Configure the Track Pan Range

The `NNS_SndPlayerSetTrackPanRange` function was added to allow the track pan range to be configured.

The track pan range is a parameter that designates the variation range for the track pan configured in the sequence data. By changing the track pan range, the variation range for the track pan configured in the sequence data can be decreased.

2.5.2.3 Addition of a Function to Obtain the Stream Data Time Length

The `NNS_SndArcStrmGetTimeLength` function was added to obtain the time length for the entire stream data.

For further details, refer to the *Function Reference Manual*.

2.5.2.4 Addition of Functions to Obtain the Sound Driver Information

Added functions to obtain information about the sound driver. These functions are listed below.

- `NNS_SndUpdateDriverInfo`
- `NNS_SndReadDriverChannelInfo`
- `NNS_SndPlayerReadDriverPlayerInfo`
- `NNS_SndPlayerReadDriverTrackInfo`

Also added the `driverInfo` demo, which displays the sound driver information on screen using these functions.

2.5.2.5 Addition of the `SND_COMMAND_IMMEDIATE` Flag for `SND_FlushCommand` Function

Made it possible to specify `SND_COMMAND_IMMEDIATE` for the argument flag in the `SND_FlushCommand` function. When `SND_COMMAND_IMMEDIATE` is designated, the issued command is processed without waiting for the next sound frame.

2.5.2.6 Added `SND_INST_NULL` to `SNDInstType`

Added `SND_INST_NULL` to the values that can be taken by the `type` member of the `SNDInstData` structure used by functions such as `SND_ReadInstData`. The instrument data configured by `SND_INST_NULL` plays no sound.

2.5.3 Bug Fixes

2.5.3.1 Release of Channels Playing Sounds with a Portamento Command

There was a bug that released all channels playing on corresponding tracks when the portamento commands `porta`, `porta_on`, `porta_off` were executed in the sequence data. This bug was fixed.

Also fixed the problem caused by this bug that caused sounds to be generated improperly when portamento commands were executed in tie mode.

2.5.3.2 Data Load Could Not be Selected in Manual Load Mode on SoundPlayer

Fixed the bug that prevented data from being loaded because the cursor could not move to the data load menu item while in the SoundPlayer manual load mode.

2.5.3.3 Unnecessary Wait Inserted at the End of the Track with `smfconv.exe`

Fixed the bug that caused an unnecessary `wait` command to be inserted at the end of each track in the `.smft` file created when converting an SMF file.

2.5.3.4 Invalid Wait Time Output in `smfconv.exe`

Fixed the bug that output an invalid `wait` time when there are multiple time signature events of the same timing in an SMF file.

2.5.3.5 False Error Occurrence in `bankconv.exe`

Fixed the bug that caused a false error when both an instrument label and an index number were specified in the bank definition file.

2.6 Changes from the 2004/12/06 Version

2.6.1 Changes to Sound Data

2.6.1.1 Addition of a Feature to Individually Load Waveform Data

Added a feature that allows the individual loading of waveform data in the waveform archive.

If multiple banks are using common waveform data, this feature allows for a simple way of using memory more efficiently without having to use the waveform data group management feature.

For further details, refer to the *Sound Archive Manual*.

2.6.1.2 Addition of a Feature that Limits the Channels a Sequence Uses

Added a feature that can limit the usable channels for a sequence on each player.

Made it possible to specify bit flags for allocatable channels in the @PLAYER section of the sound archive definition file (can be omitted).

For further details, refer to the *Sound Archive Manual*.

2.6.1.3 Support for the UNC Format Path Notation

Revisions were made so that a UNC format path notation such as //server-1/path/dir can be used in @PATH and #include.

2.6.1.4 Addition of the #include<filename> Format

Added a #include<filename> format that has the same feature as the #include "filename" in the text data file. If the #include<filename> format is used, it becomes the relative path from the sound archive definition file (*.sarc) rather than the relative path from the current file.

For further details, refer to the *Sound Tools Manual*.

2.6.1.5 Addition of the Sequence Command printvar

The sequence command `printvar` was added, which is used to debug output sequence variable values.

When a sequence is played back with NITRO-Player or SoundPlayer, the sequence variable value can be displayed on the MCS server or the output window of the IS-NITRO-DEBUGGER.

For further details, refer to the *Sequence Data Manual*.

2.6.1.6 Elimination of a Distinction between Capital and Small Letters in the Key Notation

Changes were made so that descriptions can be made in either capital or small letters for `cn4` format key notation. For example, a description such as `Cn4` or `CN4` are both possible.

2.6.2 Changes to SoundPlayer

2.6.2.1 Improvement of the Channel Meter

Changes were made so that the channel meter that had been displayed on the upper screen with SoundPlayer is displayed on the lower screen to allow the display of detailed information.

2.6.3 Changes to the Program

2.6.3.1 Addition of the `NNS_SndCaptureSetPostOutputEffectCallback` Function

Added `NNS_SndCaptureSetPostOutputEffectCallback` which registers the callback functions called as output effect postprocessors.

2.6.3.2 Addition of a Feature that Limits the Channels Used by a Sequence

Added a feature that can limit the channels that a sequence can use.

Use the `NNS_SndPlayerSetAllocatableChannel` function when configuring each player.

Use the `NNS_SndPlayerSetTrackAllocatableChannel` function when configuring each sequence playback. The specification for each track is also possible with this function.

For further details, refer to the *Sound System Manual* or the *Function Reference*.

2.6.3.3 Addition of the micThrough Demo

Added the micThrough demo which plays back microphone input in real time by using the low-level stream library `NNS_SndStrm`.

2.6.4 Bug Fixes

2.6.4.1 Fixed an SMF Converter Bug

There was a problem with an abnormal wait command being output if there were many MIDI events in one bar. This problem has been fixed.

2.6.4.2 Problem with Allocated Channels being Deallocated by `NNS_SndArcStrmStopAll`

There was a problem where allocated channels were deallocated if the `NNS_SndArcStrmStopAll` function or the `NNS_SndStopSoundAll` function was called after allocating a channel with `NNS_SndArcStrmAllocChannel`. This problem has been fixed.

2.7 Changes from the 2004/11/10 Version

2.7.1 Changes to Sound Data

2.7.1.1 Using Two Channels for Mono Stream Playback

An optional feature was added to allow the use of two channels for playing mono stream data with stereo player so that the volume becomes louder.

This feature becomes available if the option is specified when registering stream data in the sound archive definition file. For details, see the *Sound Archive Manual*.

2.7.2 Changes to the Program

2.7.2.1 Added the NNS_SndArcStrmStartEx2 Function

The `NNS_SndArcStrmStartEx2` and `NNS_SndArcStrmPrepareEx2` functions were added. With these functions, two callback functions can be registered in addition to the existing stream playback function feature.

One of the callback functions that can be registered can put multiple stream data sets together in real time and play them. This demo is in `$NitroSystem/build/demos/snd/stream-2`.

The other callback function applies effects to a specific stream playback. This demo is in `$NitroSystem/build/demos/snd/stream-3`.

For details, see the *Function Reference*.

2.7.2.2 Made the Sound Driver Thread-safe

Made all of the sound driver functions (`SND`) thread-safe.

2.7.2.3 Calling NNS_SndArcStrmAllocChannel to the Stream Player Playing Back

Made changes so that the function will fail when the `NNS_SndArcStrmAllocChannel` function was called for the stream player that is playing.

2.7.3 Changes to Manuals

2.7.3.1 Added Caution Regarding the Output Effects

Added caution regarding the output effects in *Sound System Manual*.

2.7.4 Bug Fixes

2.7.4.1 Bug that Prevented the Sound Generation of the Note After tieon

If the command is defined in the order of note command, `tieon`, and note command in the sequence data, the sound of the second note was not generated. This bug was fixed.

2.8 Changes from 2004/10/12 version

2.8.1 Changes to the Program

2.8.1.1 Resource Control Functions Added

Added a function that manages resources to avoid conflict with the NITRO-System library when using the sound driver (SND) function directly. The new functions are listed below.

- `NNS_SndLockChannel`
- `NNS_SndUnlockChannel`
- `NNS_SndLockCapture`
- `NNS_SndUnlockCapture`
- `NNS_SndAllockAlarm`
- `NNS_SndFreeAlarm`

See the *Function Reference* for details.

2.8.2 Bug Fixes

2.8.2.1 Sample of Stream Playback without Loop

When playing stream data without loops, the last several hundred samples did not play. We have fixed this bug so that all samples now play correctly.

2.8.2.2 Current Sound Archive Switching Bug During Stream Playback

The stream did not play back correctly when the Current Sound Archive was switched with the `NNS_SndArcSetCurrent` function during stream play. This bug has been fixed.

2.8.2.3 `NNS_SndArcStrmSetChannelVolume` Function Bug

There was a bug where the volume setting in the `NNS_SndArcStrmSetChannelVolume` function was not reflected correctly. This bug has been fixed.

2.8.2.4 Sequence Converter `seqconv` Command Analysis Bug

With sequence data, such as `cn4_r_if`, when attaching random and conditional assignments for note command, an error occurred during conversion. This bug has been fixed.

2.9 Changes from the 2004/09/16 Version

2.9.1 Changes to SoundPlayer

2.9.1.1 IS-AGB-MIDI Support

Real-time MIDI playback can be performed using the IS-AGB-MIDI instead of the IS-NITRO-UIC MIDI. For details of real-time MIDI playback, see *Sound Designer Guide*.

2.9.2 Changes to the Program

2.9.2.1 Added Output Effect Feature

Made it possible to apply Surround mode processing, Headphone mode processing, and Mono mode processing to the overall sound output using the `NNS_SndCaptureStartOutputEffect` function.

The Surround mode makes the sound seem more spacious than the location of the speakers when sound is output from the speakers of the Nintendo DS system. The Headphone mode reduces the load on the ears when headphones are used.

A sample demo has been added to the following location:

`$NitroSystem/build/demos/snd/outputEffect`.

See the *Function Reference* for the details about the functions.

2.9.2.2 Automated the Sleep Process

It is no longer necessary to call the functions, `:NNS_SndBeginSleep`, `NNS_SndEndSleep`, `NNS_SndStrmBeginSleep`, and `NNS_SndStrmEndSleep`, that needed to be called before and after the sleep process. These processes are performed automatically in the library.

2.9.2.3 Added the Stream Player Specification Function

Added the functions, `NNS_SndArcStrmStartEx` and `NNS_SndArcStrmPrepareEx`, for specifying the stream player number from the program.

2.9.2.4 Added the Mono Flag Setting Function

Added the `NNS_SndSetMonoFlag` function for ignoring the pan setting of each channel and producing all sounds at a central location. The `SND_SetMasterPan` and `SND_ResetMasterPan` functions were added to the sound driver to allow for the implementation of the `NNS_SndSetMonoFlag` function.

2.9.3 Changes to Sound Data

2.9.3.1 Player Label Specification

In the sound archive definition file, the player label that is defined by the @PLAYER section can be used instead of the player number that is specified by the @SEQ section. This applies only when the @PLAYER section is specified before the @SEQ section.

You can also use the stream player label in the same way as is defined by the @STRM_PLAYER section of the @STRM section.

When you want to use the player label in the @SEQ_TABLE section of a sequence archive, this can be done by getting the sound archive label file (*.sddl) using #include in the same way as a bank label is used.

Revised to use `sound_data.sarc` and `mus/se.mus` of the
`$NitroSystem/tools/nitro/SoundPlayer/data` sample. Use these as references.

2.9.4 Changes to Manuals

2.9.4.1 Added Cautions for Use when Changing to the Sleep Mode

Added cautions to the *Sound Programmer Guide* for use when changing to the Sleep mode.

2.9.4.2 Added Explanation for Not Interrupting Stream

Added an explanation to the *Sound Programmer Guide* for how to avoid interrupting stream play.

2.9.4.3 Added Introduction About Output Effects

Added an introduction to the *NITRO-Composer Overview* about output effects.

2.9.5 Bug Fixes

2.9.5.1 Infinite Loop Bug During Sequence Fadeout

Fixed the infinite loop bug that occurs when a sequence is faded out using the following functions.

- `NNS_SndPlayerStopSeqByPlayerNo`
- `NNS_SndPlayerStopSeqBySeqNo`
- `NNS_SndPlayerStopSeqBySeqArcNo`
- `NNS_SndPlayerStopSeqBySeqArcIdx`
- `NNS_SndPlayerStopSeqAll`

2.9.5.2 Mistake in Interpretation of the @PATH Command

There were mistakes in the interpretation of the @PATH command in the sound archive definition file and the bank definition file.

When #include is used to include a file in another directory and if the @path command is used in the file in the include statement, the path was interpreted as a relative path from the original directory. It was changed to interpret as a relative path from the directory file that has the included file.

Because of this fix, sound data that was converted successfully in previous conversions may have errors when they are converted. If this occurs, fix the @PATH specification so that it is the relative path from the directory where that file is.

2.9.5.3 Release Channel Process During Sequence Fadeout

Fixed the bug where the volume of the channel being released did not attenuate during the sequence fadeout.

2.9.5.4 Distinction Between Upper and Lower Case in File Names

When registering bank definition files, if you specified multiple file names that only differ in case (upper or lower), each file was registered as separate data. This has been fixed so that the bank definition files are collected as a single set of data.

2.9.5.5 Waveconv Loop Start Location Check

An error message will now be output if you attempt to convert a loop start when the loop start position is too far back to play the loop because of a hardware limitation. The threshold information has been included in the *Sound Tools Manual*.

2.10 Changes from the 2004/09/02 Version

2.10.1 Changes to Sound Data

2.10.1.1 Extension of Loop_start Sequence Command

You can now assign a value of 0 as the number of times that the `loop_start` sequence command must loop. If a value of 0 is assigned, the loop will be infinite.

2.10.1.2 Specifying Loop for Individual SMF Tracks

If MIDI control changes 89 and 90 are used, loops can be made for individual tracks. For details, see the *Sequence Data Manual*.

2.10.1.3 SMF Loop Specification

To specify all tracks to loop in SMF, markers “[” and “]” were used, but “loop_start” and “loop_end” can also be used. (The actual functionality has not changed.)

2.10.1.4 Extension of .sbd1 Files and Name Change

Previously, only bank labels were defined in `.sbd1` files. Now, labels that are defined in (`.sarc`) sound archive definition files can also be made to output.

In addition, `.sbd1` files, which were previously called “bank list files”, were changed to “sound archive label files.”

2.10.1.5 Naming of .sdl Files Unified

`.sdl` files, which were previously called “sound label lists,” “sound label list files,” and “sound label files,” are now called “sound label files.”

2.10.2 Changes to the Program

2.10.2.1 NNS_SndCaptureCreateThread Function Added

The function `NNS_SndCaptureCreateThread`, which starts up a capture thread, was added. When a capture thread is started, effect callback functions can be called from the capture thread rather than the IRQ handlers.

2.10.2.2 NNS_SndCaptureStartSampling Function Added

The function `NNS_SndCaptureStartSampling`, which samples audio output, was added. The sampled data can be obtained via a callback function.

2.10.2.3 NNS_SndArcGetSeqArcSeqCount Function Added

The function `NNS_SndArcGetSeqArcSeqCount`, which obtains the number of sequences in a sequence archive, was added.

2.10.3 Bug Fixes

2.10.3.1 NNS_SndPlayerMoveVolume Function

Fixed the bug that caused the volume to become 0 or remain unchanged when a value of 0 was specified as the change frame count in the `NNS_SndPlayerMoveVolume` function.

2.10.3.2 Unresponsive Sequences

Fixed the bug that caused the sequence that is being played to become unresponsive to control in extremely rare cases.

2.10.3.3 Incorrect Error Message in SoundPlayer

The message "Too Large Data" appeared when a sequence that has an invalid bank number is played. This message has been changed to "Invalid Bank No."

2.10.3.4 Turning Mute Off when Sound Not Muted

There was a problem that caused the sound being played to be stopped if Mute Off operation was performed when the sound was not muted, but this problem was fixed.

2.10.3.5 Bank List File Updating

Fixed the bug that updated the bank list files (`.sbd1`) that were generated by sound archiver `sndarc` even when the sound data was not modified. Now, these files are updated only when necessary.

2.10.3.6 Sequence Converter `seqconv` Error Check

Previously, no error occurred when 32 was specified as the variable number that was used with variable commands. Now, an error will be registered.

2.11 Changes from the 2004/08/10 Version

2.11.1 Real-time MIDI Playback

It is now possible to use IS-NITRO-UIC MIDI for real-time MIDI playback. Since playback is done in real time using MIDI signals, the sequence data can be checked without converting the data.

For details, see *Sound Designer Guide*.

For real-time MIDI playback, “IS-NITRO-UIC MIDI” is required.

2.11.2 Changes to Sound Data

2.11.2.1 Bank List File Output

Changes were made to output bank list file (*.sbd1) when the sound archive sndarc is executed.

The bank label that is defined in the sound archive definition file (*.sarc) is output by using #define. By using #include in the sequence archive text file and loading this file, a bank label can be used to specify the bank in @SEQ_TABLE.

Also see mus/se.mus in the \$NitroSystem/tools/nitro/SoundPlayer/data sample since the same mechanism has been applied.

2.11.2.2 Embedded SMF Text Commands

When text commands are embedded in SMF as markers, these text commands are output in the *.smft file when the file is converted. This allows you to use text commands in some parts of the file when using SMF.

For details, see the *Sequence Data Manual*.

2.11.2.3 Multiple Definition of the Same Label in Sequence Archives

Changes were made so that an error does not occur if the index numbers are the same when the same label is defined in different archives.

2.11.3 Changes to the Program

2.11.3.1 Feature for Awakening from the Sleep Mode

Before and after the PM_GoSleepMode function that changes to the Sleep mode, call the NNS_SndBeginSleep and NNS_SndEndSleep functions to allow the system to return to the previous state when it awakes from the Sleep mode.

2.11.3.2 Clearing the Sound Heap

The sound archive management region and the sound heap region that includes the stream buffer can be released safely. To make the regions usable again, it is necessary to call various setup functions such as the NNS_SndArcSetup and NNS_SndArcStrmSetupPlayer functions.

2.11.3.3 Added the NNS_SndArcSetup Function

Added the `NNS_SndArcSetup` function, which makes the sound archive that has released the sound archive management region usable again.

2.11.3.4 Added the NNS_SndArcStrmSetupPlayer Function

Added the `NNS_SndArcStrmSetupPlayer` function that makes the stream player that released the stream buffer usable again.

2.11.4 Bug Fixes

2.11.4.1 Player Heap Release Bug

There was a bug that caused problems when releasing the region in the sound heap in which the player heap was allocated. This bug was fixed.

2.11.4.2 Track Output Error of SMF Converter `smfconv`

Fixed the bug that output only the data in the first track when using one MIDI channel on multiple tracks.

Also fixed the bug that output the data in the track that was not being used.

2.11.4.3 Shifting the Loop Starting Point in SMF Converter `smfconv`

Fixed the bug that caused the loop starting point to shift to the location of the note command when the loop start point was placed before the note command.

2.11.4.4 Path Interpretation Error of Bank Converter `bankconv`

Fixed the bug that caused an error when `@WAVEARC` path was `@PATH ". "` in the sound archive definition file `*.sarc`.

2.11.4.5 #define Multiple Definition Error

When the same symbol was defined twice with `#define` in text data files such as sound archive definition file and bank definition file, it became a multiple definition error even when the content was exactly the same. Changes were made so that it will be ignored if the contents are the same.

2.12 Changes from the 2004/07/20 Version

2.12.1 Support for Stream Playback

Stream playback is now possible.

For details on features, see the *Sound System Manual*.

2.12.1.1 Changes to the Sound Archive Definition Ffile

The stream data section, which registers the stream data, and the stream player information section, which registers the stream player, were added.

For details, see the *Sound Designer Guide* and the *Sound Archive Manual*.

2.12.1.2 Changes to the SoundPlayer

The SoundPlayer can now play stream data.

For details, see the *Sound Designer Guide*.

2.12.1.3 Sound Archive Stream Library Added

The Sound Archive Stream Library, which plays stream data, was added. This is a group of functions that start with `NNS_SndArcStrm`.

For details, see the *Sound Programmer Guide* and the *Function Reference*.

2.12.2 Changes to the Program

2.12.2.1 Stream Library Added

The Stream Library, which is a lower-level library than the Sound Archive Stream Library, was added. This is a group of functions that start with `NNS_SndStrm`.

For details, see the *Function Reference*.

2.12.2.2 NNS_SndStopSoundAll Function Added

The function `NNS_SndStopSoundAll`, which stops all sounds, was added. Before transitioning to sleep mode or turning off power to the sound circuitry, make sure to stop all sounds with this function.

2.12.2.3 NNS_SndHeapDestroy Function Added

The function `NNS_SndHeapDestroy`, which destroys the sound heap, was added.

2.12.3 Changes to Sound Data

2.12.3.1 Changes to SMF Converter `smfconv` Output Label Names

The function `smfconv` now adds the SMF file name to the beginning of the labels used in the `.smft` file.

For example, the file `sample.smft`, which results from converting `sample.mid`, previously contained the label `Track_0`. But it now contains the label `SMF_sample_Track_0`.

2.12.4 Bug Fixes

2.12.4.1 Sound Archiver `sndarc` Abnormal Termination

Previously, the sound archiver would terminate with an error when conversion was attempted using a sound archive definition file in which registration numbers of sequence data, and so forth, were specified directly and some numbers contained no data. This bug is fixed.

2.12.4.2 Bank Converter `bankconv` Abnormality Error

There was a bug that caused an error during bank conversion due to a mistake in the path interpretation when the directory trees of bank files and waveform list files are in certain state. This bug was fixed.

2.12.4.3 Irregular Order of SMF Converter `smfconv` Meta Event

If multiple meta-events occurred during the same tick, the output order would become irregular. This was corrected so that the output is now in the order stored in SMF.

2.13 Changes from the 2004/06/10 Version

2.13.1 Changes to the Binary Data Format

The binary format for the sound data has changed. Now, you must reconvert the sound data with `ReMakeSound.bat`.

The bank data has grown slightly in size. If there is not enough memory, the data load may fail.

2.13.2 Changes to SoundPlayer

2.13.2.1 Name of Executable File has Changed

The name of the executable file has changed from `SoundPlayer.bin` to `SoundPlayer.srl`.

2.13.2.2 Changes in the Default Platform

Previously, the TEG executable file was created by default. Now, a TS executable file is created by default. To create a TEG executable file, set the `NITRO_PLATFORM` environment variable to `TEG`.

2.13.2.3 Sequence Type Selection Feature

Previously, no more than two sequences and two sequence archives could be played at the same time. Now, you can select between playing any combination of sequences and sequence archives. For example, you can play up to four sequences simultaneously.

To select the sequence type, move the +Control Pad up and down while pressing the L Button.

2.13.3 Changes to the Program

2.13.3.1 Waveform Playback Library Added

Added a waveform playback library with function names that start with `NNS_SndWaveOut`. This waveform playback library allows you to directly play waveform data that is in memory, such as samples taken with the microphone.

The demo program is in `$NitroSystem/build/demos/snd/waveout`.

2.13.3.2 Effects Features Added

Added effects functionality that uses sound capture. By processing effects using a callback function, you can create effects such as low pass filter can be applied on all sounds.

The demo program is in `$NitroSystem/build/demos/snd/effect`.

2.13.3.3 Changes in the Arguments for `NNS_SndCaptureStartReverb`

The type of the argument `format` of the `NNS_SndCaptureStartReverb` function was changed from `SNDCaptureFormat` to `NNSSndCaptureFormat`.

For details, see the *Function Reference*.

2.13.3.4 Changes in the `NNSSndArcBankInfo` Structure

`waveArcNo`, a member of the `NNSSndArcBankInfo` structure, was changed from `u16` to `u16[4]` type. The previous value is stored in `waveArcNo[0]`.

2.13.3.5 Functions Deleted

The following deprecated functions were deleted.

- `NNS_SndArcPlayerInit`
- `NNS_SndArcPlayerStartSeqWithPlayerNo`
- `NNS_SndArcPlayerStartSeqArcWithPlayerNo`
- `NNS_SndPlayerStopSeqByNumber`
- `NNS_SndPlayerPauseByNumber`
- `NNS_SndPlayerCountPlayingSeqByNumber`
- `NNS_SndHandleReleasePlayer`

2.13.4 Changes to Sound Data

2.13.4.1 Ability to Use Multiple Waveform Archives per Bank Added

You can now associate multiple waveform archives (a maximum of four) with a single bank. For details, see the *Bank Data Manual* or the *Sound Archive Manual*.

2.13.4.2 Option to Delete Sound Archive Symbol Data

If you use option `-b` with the `sndarc` sound archiver, the sound archive that is created will not include symbol data.

2.13.5 Bug Fixes

2.13.5.1 Mistake in the Value for the Maximum Number of Simultaneous Sequence Playback

Fixed the problem that caused an error when the value for the maximum number of simultaneous sequence playback in the sound archive definition file was specified as 16.

2.14 Changes from the 2004/04/12 Version

2.14.1 Migration from Previous Environment

Copy the `MakeSound.bat` and `ReMakeSound.bat` files in the `$NitroSystem/tools/nitro/SoundPlayer/data` directory to the directory with the `sound_data.sarc` file. Then double click on `ReMakeSound.bat` to execute.

This will create `sound_data.o`. Link it to the program as before. Thereafter, it can be used as it was in previous versions.

However, this version offers support for a file system. Sound data management using the file system is recommended.

2.14.2 File System Support

Rather than linking to the program, sound data can now be stored in ROM and loaded.

2.14.2.1 Sound Archive Files

The `sound_data.o` file is no longer linked. Instead `sound_data.sdat` is stored in ROM. Sound designers must provide programmers with a `sound_data.sdat` file rather than a `sound_data.o` file. Note that the `sound_data.sdat` format has changed. Be sure to use a properly converted `sound_data.sdat` file.

2.14.2.2 Embedding Sound Data

The programmer must store the `sound_data.sdat` file in ROM. For details, see the *Sound Programmer Guide*.

2.14.2.3 Initialization of the Sound Library

The sound library initialization process has changed. For details, see the *Sound Programmer Guide*.

2.14.2.4 Memory Management

Memory management is required for sound data. For details, see the *Sound System Manual*, the *Sound Designer Guide*, and the *Sound Programmer Guide*.

2.14.2.5 ARM7 Component Changes

The name of the ARM7 component was changed from "prototype" to "ferret." For details, see the *Sound Programmer Guide*.

2.14.3 Changes to SoundPlayer

2.14.3.1 Changes to the Convert Command

Previously, a make command was used for conversion. Now `MakeSound.bat` is used.

`MakeSound.bat` can be executed by double clicking it from Windows Explorer.

To reconvert, use `ReMakeSound.bat` instead of `make rebuild`.

2.14.3.2 Label Name Display

Sequence label names can now be displayed on the SoundPlayer screen. When the X Button is pressed, the parameters will be displayed as in the previous version.

2.14.3.3 Manual Loading Mode Added

A manual loading mode was added to allow you to confirm the data size. For details, see the *Sound Programming Guide*.

2.14.4 Changes to Player

2.14.4.1 Playback of Multiple Sequences

A single player can now play multiple sequence files. The number of sequences that can be played simultaneously can be set individually for each player. (The default is 1.)

2.14.4.2 Number of Players

The number of players was increased from 16 to 32. However, the number of sequences that can be played simultaneously by the system is still 16.

2.14.4.3 Specifying the Number of Sequences that Can Be Played Simultaneously

The sound archive can specify the number of sequences that can be played simultaneously on each player.

```
@PLAYER
3 : 3      ; Player 3 can play up to three sequences
4 : 2      ; Player 4 can play up to two
```

If no number is specified, the default value is one.

Also, the number can be set in the program using the `NNS_SndPlayerSetPlayableSeqCount` function.

2.14.4.4 Player Priority During Fadeout

Player priority was changed to 0 (lowest) during fadeout.

2.14.5 Changes to Sound Data

2.14.5.1 Slight Changes to Release/Decay

The release and decay of the envelope were slightly modified so that faster release or decay is possible. Be aware that noise can be generated due to the rapid release if the release value is set to 127.

The relationship between the previous values and the new values is as follows.

Previous Value	Correspondence with New Value
51- 127	Equivalent to 49-125.
47 - 50	Subtract 1 or 2 from each.
0 - 46	No change.

2.14.5.2 Sound Map File Output

A sound map file is now saved when a sound archive is created. A sound map file is a text file that contains information about the data in the sound archive, such as the sound data size.

The sound map file is in the same directory as the sound archive definition file, but has the `.smap` extension.

2.14.5.3 Specifying Index Numbers

Sequence and bank numbers in the sound archive were previously assigned in order. Now, you can assign them yourself. Also, you can now assign index numbers to sequence tables for sequence archives. The format is as follows.

* Specify only a label (the previous one)

```
SEQ_TITLE_BGM : SMF, "title.mid" , ...
```

* Specify both a label and a number

```
SEQ_TITLE_BGM = 2 : SMF, "title.mid" , ...
```

* specify only a number

```
2 : SMF, "title.mid" , ...
```

Previously each instrument in the bank definition file was assigned a number, but now you can also assign labels using the syntax above. If you assign labels, you can use the program number list file, which is explained below.

2.14.5.4 Outputting the Program Number List

If you assign labels to the program numbers in the bank definition file, you can use the program number list file. This file allows you to replace program numbers with labels. An example is shown below.

```
#define PRG_PIANO      0
#define PRG_GUITAR     1
#define PRG_HARP       2
```

This file is found in the same directory as the bank definition file, but uses the extension `.spd1`. By including this file in the text sequence archive, labels can be used instead of program numbers.

```
#include "../bnk/se.spd1"

test:
    prg PRG_PIANO
    cn4 64, 48
    fin
```

2.14.5.5 Expression Added

MIDI control change 11 ("expression") was converted to the `volume2` command. The `volume2` command functions just like the `volume` command, but generates its own individual effects.

2.14.5.6 Sweep_pitch Can Be Used from MIDI

Previously, MIDI control change 28 was used for the `sweep_pitch` command, but MIDI control change 29 was converted to the `sweep_pitch` command, with 24 times its previous value. Using this control change, the pitch can be changed \pm two octaves.

2.14.5.7 Maximum Value for `prg` and `mod_delay` Commands Changed

The maximum value for the sequence commands `prg` and `mod_delay` was reduced from 65,535 to 32,767.

2.14.6 Addition of Extended Sequence Commands

2.14.6.1 Random Commands

You can now specify random numbers in some sequence commands.

```
pitchbend_r -12, 12
pan_r 64 - 32, 64 + 32
```

Once the minimum and maximum values are specified, a random value will be set within that range.

The commands that permit the use of random numbers are listed in the *Sequence Data Manual*.

2.14.6.2 Variable Commands

Variables can now be manipulated in a sequence. Using variables, you can easily create a sequence that rises in pitch with each iteration.

For details, see the *Sequence Data Manual*.

2.14.6.3 Control Commands

Using variables, you can now control whether or not a sequence command is executed.

For details, see the *Sequence Data Manual*.

2.14.7 Changes to the Sound Program

2.14.7.1 Deletion of the NNS_SndArcPlayerInit Function

The `NNS_SndArcPlayerInit` function will be deleted. See the *Function Reference* for functions that replace it.

This function remains for now, but it will be completely deleted in the next release.

2.14.7.2 Deletion of the NNS_SndArcPlayerStartSeqWithPlayerNo Function

The `NNS_SndArcPlayerStartSeqEx` and `NNS_SndArcPlayerStartSeqArcEx` functions were added. Since these functions can replace the `NNS_SndArcPlayerStartSeqWithPlayerNo` function and `NNS_SndArcPlayerStartSeqArcWithPlayerNo` function, the latter two functions will be deleted.

These functions remain for now, but they will be completely deleted in the next release.

2.14.7.3 Function Name Changes

The names of the following functions were changed. The old function names remain, but they will be completely deleted in the next release.

Deprecated Function	New Function Name
<code>NNS_SndHandleReleasePlayer</code>	<code>NNS_SndHandleReleaseSeq</code>
<code>NNS_SndPlayerStopSeqByNumber</code>	<code>NNS_SndPlayerStopSeqByPlayerNo</code>
<code>NNS_SndPlayerPauseSeqByNumber</code>	<code>NNS_SndPlayerPauseSeqByPlayerNo</code>
<code>NNS_SndPlayerCountPlayingSeqByNumber</code>	<code>NNS_SndPlayerCountPlayingSeqByPlayerNo</code>

2.14.7.4 Sequence Variable Functions Added

`NNS_SndPlayerReadVariable()` and `NNS_SndPlayerWriteVariable()` can be used to read the variables set by the sequence data or to write to the variable in the sequence data. With these functions, you can view the status of a sequence being played back and alter its processing flow.

2.14.7.5 Obtaining the Number of Ticks

With `NNS_SndPlayerGetTick()`, you can view the tick count of the sequence being played.

2.14.7.6 Added Other New Functions

Many other functions were added. For details, see the *Function Reference*.

2.14.8 Bug Fixes

2.14.8.1 Voice Priority

There was a problem that caused the voice priority setting in the sound data to be reflected incorrectly. This problem was fixed.

2.14.8.2 The NNS_SndPlayerSetTrackPan Function

Previously, an assert was generated when the pan argument was from -64 to 63. An assert is now generated when the value is from -128 to 127, as the specification requires.

2.14.8.3 Release Problem with NNS_SndPlayerStopSeq

A bug caused sequences with long release values to stop abruptly when halted from the application. This bug is fixed.

2.15 Changes from the 2004/04/07 Version

2.15.1 Changes to Sound Data

2.15.1.1 Sequence Label

Previously with the sequence label name of the sequence archive, the same label name could be given to different sequence archives. A change was made so an error occurs if this is done.

2.15.1.2 Preprocessor Directives

A change was made so a preprocessor directives can be used for the description of the sound data text file. Preprocessor directives are a group of commands that support text description. For details, see the *Sound Tools Manual*.

2.15.2 Changes to the Programs

2.15.2.1 Addition of a Track Mute Function

The `NNS_SndPlayerSetTrackMute()` function was added. This allows a mute for each track of a sequence.

2.15.3 Changes to the Manuals

2.15.3.1 Sound Tools Manual

The *Sound Tools Manual* was created. This is a supplemental explanation of the sound tools. The file is `$NitroSystem/docs/NitroComposer/NITRO_Composer_SoundToolManual.pdf`.

In addition, the explanation about how to use the converter in another manual was moved to the *Sound Tools Manual*.

2.15.3.2 Overview

“How to read through the manuals” was added to the *NITRO-Composer Overview*.

2.15.3.3 Fixing Errors

Added methods for fixing general errors when converting data to the *Sound Designer Guide*.

2.15.4 Bug Fixes

2.15.4.1 Function for Setting Up Track Parameters

There was a bug in which the track parameter setting functions such as `NNS_SndPlayerSetTrackVolume()` were not working properly. This bug was fixed.

2.16 Changes from the 2004/03/18 Version

2.16.1 Installing the NITRO-SDK Patch

A patch is required for the NITRO-SDK. For details, see the NITRO-System documentation.

2.16.2 Changes to SoundPlayer

2.16.2.1 Directory Changes

The `$NitroSystem/build/tools/nitro/SoundPlayer` directory is no longer present. Instead, the sample data and the build environment for SoundPlayer is in `$NitroSystem/tools/nitro/SoundPlayer/data` directory.

2.16.2.2 Executable File

The executable file generated by `make` was changed from `$SoundPlayer/bin/ARM9-TEG/Release/main.bin` to `$data/SoundPlayer.bin`.

2.16.2.3 Migration from Previous Environment

Remove `$SoundPlayer/data` directory from `$SoundPlayer` directory. Copy the makefile from `$NitroSystem/tools/nitro/SoundPlayer/data` and overwrite the makefile in the `data` directory. After this, a `make` in the `data` directory allows it to be used the same as before.

However, in order to handle the binary format described below, do a `make rebuild` in the `data` directory only in the very beginning.

2.16.3 Changes to the Sound Archive

2.16.3.1 Sound Label List File

Changes were made so a sound label list file with the extension `*.sادل` is output at the same time as the sound archive is created. This file contains the serial numbers of the sequence data, and so forth, defined as labels using `#define`. The sound programmer can use labels instead of numbers to specify sequence data by placing this file in an include statement in the source file.

2.16.4 Changes to Sequences

2.16.4.1 Changes to Binary Format

The sequence binary format was changed. The sequence data needs to be converted again, but the previous data can be used tentatively as is.

2.16.4.2 Addition of Envelope Command

A sequence command was added so an envelope value can be set in the sequence. It can be used with MIDI as well.

2.16.4.3 Addition of Loop Command

A sequence command was added to repeat a section a specified number of times.

2.16.4.4 porta Command from MIDI

Made changes to convert MIDI control change (84) to the `porta` command.

2.16.5 Changes to Manuals

2.16.5.1 Sound System Manual

Created the *Sound System Manual*. The file is

`$NitroSystem/docs/NitroComposer/NITRO_Composer_SoundSystemManual.pdf`.

2.16.5.2 Reference Manual

Created a *Function Reference* manual. View it under `$NitroSystem/man/en_US/index.html`.

2.16.5.3 Sound Programmer Guide

An overall change to the organization was made. An explanation of the library organization was added.

2.16.5.4 Envelope Specifications

Detailed specifications about the envelope values were added to the *Bank Data Manual*.

2.16.5.5 Waveform File Format

Detailed specifications about the waveform file format were added to the *Bank Data Manual*.

2.16.5.6 Pitch Notation

A description about notations such as for a flat was added to the explanation of pitch notation in the *Sound Designer Guide*.

2.16.5.7 Voice Priority

The *Sound Archive Manual* and the *Sequence Data Manual* stated that “when the voice priority is the same, the one that arrived later is given priority.” This statement was changed to “when the voice priority is the same, it is indeterminate which one will be given priority.”

2.16.5.8 Information about SMF Compatibility with MusicPlayer2000

Information about SMF compatibility with MusicPlayer2000 was added to the *Sequence Data Manual*.

2.16.6 Changes to Programs

2.16.6.1 Addition of Functions

Various functions were added. For details, see the *Function Reference*.

2.16.6.2 Sample Program

A sample program was created. It is in the directory `$NitroSystem/build/demo/snd`.

2.16.7 Bug Fixes

2.16.7.1 waveconv Error

A bug was found in which an error like the following appeared for a particular waveform file and conversion could not be made. This bug is fixed.

```
Assertion failed: (loopen % align) == 0, file .\waveconv.cpp, line 540
```

2.16.7.2 seqconv Error

A bug was found in which conversion of extremely large sequence files was attempted, but an error occurred and the conversion was not possible. This bug is fixed.

2.16.7.3 Checking Multiple Definition of Label

An error did not occur even if a label was defined twice in the sound archive and sequence archive. This was fixed so that an error occurs.

2.16.7.4 Transpose

A bug was found in which an error occurred and conversion could not be done when a negative value was specified with the sequence command `transpose`. This bug is fixed.

2.16.7.5 Sequence End During Sound Generation

A bug was found in which a release is not done with the set value if `fin` is executed during sound generation in the sequence. This bug is fixed.

2.16.7.6 Player Number

A bug was found in which an error did not occur even if 16 or more player numbers were specified for the player number, but unexpected problems occurred during execution. A change was made so an error is output during conversion.

2.16.7.7 SMF Conversion with sndarc

A bug was found with the sound archive definition file. If the *.smft file revision date and time was more recent than the SMF, the content of the *.smft file was reflected even if SMF was specified for the file type. This bug was fixed. If you wish to use the content of the *.smft file, set the file type to TEXT.

2.16.7.8 Bank Convert with sndarc

A bug was found in which an incorrect sound was played back if the bank and waveform archive association was changed with the sound archive definition file, and the output file was not deleted once. This was fixed.

2.17 Changes from the 2004/03/02 Version

2.17.1 Changes to the Binary Data Format

The binary data format was changed. As a result of this change, all of the sound data must be converted again.

If a `make rebuild` was executed in the `$SoundPlayer/data` directory, all of the data will be converted again. However, fixes associated with the changes to the text sequence format described below must be done prior to this.

If the sound data cannot be rebuilt immediately, use the sound data described below temporarily. Sounds will not be played, but unpredictable problems such as the program running out of control can be avoided.

The sound data described above is in the `$NitroSystem/tools/bin/` directory.
`null_sound_data.o` or `null_sound_data.sdat`.

2.17.2 Changes to the Text Sequence Format

The text sequence format was changed. The sequence file created with the text editor needs to be revised. The major changes are described below. For details, refer to the *Sequence Data Manual* (`NITRO_Composer_SoundSequenceManual.pdf`).

If the standard MIDI file is reconverted once it can be used without any revisions.

2.17.2.1 Comments

The character for starting a comment was changed from a number sign (#) to a semi-colon (;). Make the appropriate changes.

2.17.2.2 Local Labels

Changes were made so labels starting with an underscore (`_`) are handled as local labels. Be careful if labels beginning with an underscore are being used.

Local labels are labels valid only between normal labels. If they are between different labels, a local label with the same name can be used.

2.17.2.3 Changes to Sequence Archive Header Section

Changed the format of the sequence archive header section.

Code 2-1 Prior to Change

```
format_ssar

sound_count _table_start, _table_end
_table_start:

snd note_only,      0,  65, 96, 64, 10
snd loop_seq,       0,  55, 96, 64, 10
snd call_seq,       0,  55, 96, 64, 10
snd porta_seq,      0,  65, 96, 64, 10
snd mod_seq,        0,  65, 96, 64, 10
snd tie_seq,        0,  65, 96, 64, 10
snd waitoff_seq,    0,  65, 96, 64, 11
snd opentrack_seq,  0,  65, 96, 64, 11

_table_end:
```

Code 2-2 After Change`@SEQ_TABLE`

<code>SE_COIN:</code>	<code>note_only,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_AMBULANCE:</code>	<code>loop_seq,</code>	<code>0,</code>	<code>55,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_PATTERN:</code>	<code>call_seq,</code>	<code>0,</code>	<code>55,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_PORTAMENT:</code>	<code>porta_seq,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_VIBRATE:</code>	<code>mod_seq,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_VIBRATE2:</code>	<code>tie_seq,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>10</code>
<code>SE_SUPER_MARIO:</code>	<code>waitoff_seq,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>11</code>
<code>SE_SUPER_MARIO2:</code>	<code>opentrack_seq,</code>	<code>0,</code>	<code>65,</code>	<code>96,</code>	<code>64,</code>	<code>11</code>

`@SEQ_DATA`

2.17.3 Changes to the Sequence Commands

2.17.3.1 opentrack Command

When using the sequence command `opentrack`, a track must now be secured in advance using the newly created sequence command `alloctrack`.

2.17.3.2 Note Command

Previously, the sequence did not stop if 0 was specified for the note length in note wait mode. This was changed so that the sequence waits until the waveform data playback completes.

2.17.3.3 Portamento

The sequence commands `porta_time` and `porta_on` were added to provide finer control of portamento. Changes were also made so these commands can be used from SMF.

2.17.3.4 Sweep

Added the sweep command `sweep_pitch`.

2.17.4 Bug Fixes

2.17.4.1 SMF Converter `smfconv`

Fixed a bug in which an incorrect conversion was made if the time signature was changed during a song.

2.17.4.2 Sequence Converter `seqconv`

Fixed the problem in which an error did not occur if an invalid label was specified.

2.17.4.3 Sequence Playback Immediately After Startup

A bug in previous versions was found in which sequence playback immediately after startup sometimes failed. This bug is fixed.

2.18 Changes from the 2004/02/17 Version

2.18.1 Addition of Bank Instrument Type swav

This is the same as the instrument type PCM that was eliminated previously. It is used when specifying a *.swav file as a waveform file, not AIFF or WAV.

```
0 : SWAV, "piano.swav", cn4, 127, 127, 127, 120
```

2.18.2 Changes to *.swav Filenames Generated from Bank

Previously only the extension of the filename of the *.swav file generated when the instruction type of PCM16, PCM8, or ADPCM was specified was changed to *.swav. This was changed to filenames containing format information. See examples below.

- With PCM16, .pcm16.swav
- With PCM8, .pcm8.swav
- With ADPCM, .adpcm.swav

For example, with:

```
0 : ADPCM, "piano.aiff", cn4, 127, 127, 127, 120
```

piano.adpcm.swav is generated.

2.18.3 Addition of Modulation Range

Added a command to setup a modulation range in the sequence command.

```
mod_range range
```

The maximum variation width of vibrato can be changed from the default of ± 1 semi-tone.

2.18.4 MIDI Control Code Allocation

MIDI control codes were newly assigned to the following sequence commands.

main_volume	12
Transpose	13
mod_range	23
mod_delay	27

"27" was added to the 26 of mod_delay already present. With 27 the value is increased by 10 times and output. prio(33) was changed to control code 14.

2.18.5 Changes to the Sweep Command

The sequence command `sweep` was eliminated. The `porta` command is used instead. In addition, `sweepend` was changed to `porta_off`. Both can be used in the same way.

2.18.6 Adjustment of the Envelope

If decay and release were set to the value in the range of 0 to 45, the decay and release will become even slower than before.

2.18.7 Playback of Multiple Sequence Archives

Changes were made so the second sequence archive and on can be played back with the SoundPlayer.

2.18.8 Bug Fix of Runtime Error when Sound Archive Section Empty

Fixed a bug in which a runtime error occurred and the program stopped if the `@SEQARC` section of the sound archive definition file was empty.

2.18.9 Bug Fix of Player Priority Setup for the Sequence

Fixed a bug in which the player priority set in the `@SEQ` section of the sound archive definition file was set incorrectly.

2.19 Changes from NITRO-Composer Specification Proposal

2.19.1 Specification Changes for Priority Setup

- Changes were made so that the player priority and the voice priority could be set up in the sequence of the sound archive
- Eliminated the setting of the voice priority of the sequence archives in the sound archive
- Changes were made so that the player priority and the voice priority could be set in the sequence table of the sequence archive

In other words, the parameters set up in the sequence of the sound archive and those set in the sequence table of the sound archive were made to be the same.

2.19.2 Elimination of Bank Instrument Type PCM

In the bank file, descriptions like the following are no longer possible.

```
0 : PCM, "piano.swav", cn4, 127, 127, 127, 120
```

Instead of the above, the following is used.

```
0 : ADPCM, "piano.aiff", cn4, 127, 127, 127, 120
```

PCM16 or PCM8 can be used in place of ADPCM.

In summary, for the waveform file, only AIFF or WAV can be specified and not *.swav. This measure was taken to avoid confusion. If a solution is found, a change will be made so *.swav can be specified again.

3 Known Issues

There are currently no known issues.

© 2005-2007 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed or loaned in whole or in part without the prior approval of Nintendo.