NintendoWare for CTR

Layout Binary Converter

2010/12/07

Version 1.2.3

PROVISIONAL TRANSLATION

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

Table of Contents

Tables

1 Introduction

The Layout Binary Converter is a command line tool for converting layout data files into layout binary files that the Layout library uses as resources. This document describes how to use the Layout binary converter and its features.

NintendoWare for CTR is referred to simply as NintendoWare throughout this document.

2 Operating Environment

Microsoft .NET Framework 3.5 Service Pack 1 is required to execute the Layout Binary Converter. Operations cannot be guaranteed for any other version of the Microsoft .NET Framework.

3 Using the Layout Binary Converter

3.1 Executing the Layout Binary Converter

The two ways to execute the Layout Binary Converter are:

- From the command line (with the target of the command being a LayoutEditor save file, hereafter referred to as a layout data file)
- From inside LayoutEditor

3.1.1 Executing from the Command Line

The two forms of syntax for executing the Layout Binary Converter from the command line are:

- `NW4C_LayoutConverter.exe [options] <input_layout_data_files> <output_directory>`
- `NW4C_LayoutConverter.exe [options] <input_directory> <output_directory>`

For example:

```
NW4C_LayoutConverter.exe  layout.clyt  animation.clan  outDir
NW4C_LayoutConverter.exe  -u  -g  inDir  outDir
```

More than one input layout data file may be specified. If a directory is specified for input, all layout data files found in the specified directory are searched and used as input layout data files. If layout data and animation data are located in the same directory, binary conversion can be performed on all files at once by specifying the directory without having to specify each file separately.

3.1.2 Executing from Inside LayoutEditor

Layout data and animation data created using LayoutEditor can be saved directly to layout binary files from LayoutEditor. For details, see the documentation on LayoutEditor.

3.2 Layout Binary Converter Options

Options used with `NW4C_LayoutConverter.exe` are specified with a hyphen followed by an alphabetic character. When specifying two or more options simultaneously (such as `-u` and `-g`), the options must be specified separately, for example, `-u -g` rather than combined as `-ug`.

Available options are listed in Table 3-1. See Chapter 4 Features of the Layout Binary Converter for details about each option.

Table 3-1 Layout Binary Converter Options

Option	Description
`-h`	Displays a help message and exits.
`-u`	Updates only those files whose contents have changed.
`-g`	Outputs binary files for the animation at the animation section tag level.
`--no-convert-cvtrchar`	Does not convert any hexadecimal text (which begins with ^ followed by the numbers 00 to 1F) in the text box to the text codes 00 to 1F.
`--omit-samekey`	This option can be used if the `-g` option is specified. A key is not output if the value does not change within the frame area specified by the animation section tag, excluding the first animation section tag.
`--omit-samekey-all`	This option is valid when the `-g` option is specified. A key is not output if the value does not change within the frame area specified by the animation section tag.
`--omit-nokey-all`	This option is valid when the `-g` option is specified. Animation data are not output if there are no keys within the frame area specified by the animation section tag.
`--bake-infinity`	Repeats key output to recreate the curve when the area outside of the key frame segment is not a constant.
`--no-taginfo`	The file will not include animation section tag information. This is effective when outputting in animation section tag units.
`--no-check-version`	The version of the layout file is not checked. When this option is specified, the layout library may not operate correctly, depending on the file version.
`--cvtr-ref-tex-only`	Processes only the texture files actually used for material or texture pattern animation.

4 Features of the Layout Binary Converter

4.1 Conversion to Layout Binary Files

The relationship between the file extension for layout data files and the file extension for layout binary files is shown in Table 4-1.

Table 4-1 Relationship of File Extensions

File Extension of the Layout Binary File Created After Conversion	File Extension of the Layout Data File To Be Converted Using the Layout Binary Converter
`bclyt`	`clyt`
`bclan`	`clan`

4.2 Resource Types

The Layout Binary Converter creates a directory for each type of resource in the output directory. These directories are used to store layout files after binary conversion, as well as any font and texture image data used in the layout.

Table 4-2 Resource Types and Directories Where Stored

Resource Type	File Extension	Directory Where Stored
Layout data	`bclyt`	`blyt`
Animation data	`bclan`	`anim`
Font resource	`bcfnt`	`font`
Archive font resource	`bcfna`	`fnta`
Texture image	`bclim`	`timg`

The Layout library gets resources according to the directory names listed in Table 4-2. For this reason, converted files must be stored in the appropriate directory based on the type of file.

Files with the file extension `bclim` are image files that have been converted from TGA files, which must be NW4C TGA files (ones with additional information). The texture image included in the additional information is used.

4.3 Storing Names in Resources

Just as with layout data files, names for both panes and groups are stored in layout binary files. The Layout library looks at the name when searching for data and applying the data to animations.

Resource names, such as those used for textures, must be unique within the layout binary file. Because converted files are saved under each resource type, only the last file converted is kept in the case of resource files that have the same name and type. This is true even if the source locations for the conversion are different.

4.4 Binary Conversion of Animation Data

4.4.1 Data Format of Animations

When a layout data file is converted to a layout binary file, the data format used to store animations differs depending on the type of animation.

4.4.2 Step Format

This format stores key data as a (frame, value) pair. Frame is stored as a single-precision floating point number; value is stored as an unsigned 16-bit integer.

4.4.3 Hermitian Interpolation Format

This format stores key data as a (frame, value, slope) triple.

Values for frame, value, and slope are all stored as single-precision floating point numbers.

4.4.4 Animation Types and Their Data Formats

Depending on the animation type, animation data is stored in either step format or Hermitian interpolation format.

Table 4-3 Animation Types and Their Data Formats

Animation Type	Data Format
Pane SRT animation	Hermitian interpolation format
Visibility animation	Step format
Vertex color animation	Hermitian interpolation format
Material color animation	Hermitian interpolation format
Texture SRT animation	Hermitian interpolation format
Texture pattern animation	Step format

4.5 File Conversion Options

When the `-u` option is specified, a binary comparison is made between the converted file and the existing file. Files are not updated if the contents are the same. This allows you to prevent the time stamp of files from being updated unnecessarily.

When the `-g` option is specified, an animation binary file is created for each tag located in the animation. This option can be used to extract any given segment from a serial animation.

When the `--no-convert-cvtrchar` option is not specified, any two-digit hexadecimal text (beginning with ^ followed by the numbers 00 to 1F) included in the text box text is converted to the text codes 00 to 1F. This conversion does not occur when this option is specified.

The `--omit-samekey, --omit-samekey-all` and --omit-nokey-all options are available when the `-g` option is also specified. If even a single key is set in the frame area, one key is typically output even if the value does not change within the frame area specified by the animation section tag. When the --omit-samekey option or the --omit-samekey-all option is specified, a key will not be output if the value does not change within the frame area specified by the animation section tag. However, for the `--omit-samekey` option, the key of the first animation section tag of all the animation section tags with **output binary** checked is always output. If the --omit-nokey-all option has been specified, no animation data will be output if there are no keys set within the frame area specified by the animation section tag. When this is the case, any changes to values are ignored. If the animation is not being switched in the middle and the animation section tag frame count's worth of animation is always played, using these options reduces the size of the animation binary.

`--bake-infinity` repeats key output to recreate the curve when the area outside of the key frame segment is not a constant. Be aware that the size of the output animation binary file increases with the number of repeats.

`--no-taginfo` is a valid option when the `–g` option is specified. Normally, when outputting binary files for each animation section tag, information related to the animation section tag is included in the binary file. When this option is specified, the binary file does not include the animation section tag information.

When `--no-check-version` is specified, the version of the layout file is not checked. Normally, the layout file version is checked, and if the binary converter does not recognize the version, the process is aborted. When this option is specified, the layout library may not operate correctly, depending on the file version.

When `--cvtr-ref-tex-only` is specified, of the texture files stored in the layout file, only the texture files that are actually used for material or texture pattern animation are converted. If this option is not specified, all the texture files stored in the layout file (clyt) are converted. If a texture pattern animation is used in the clan file when this option is specified, a clyt file with the same name must be placed in the same folder with the clan file.

5 Revision History

Version	Revision Date	Category	Description
1.2.3	2010/12/07	Changed	Updated document properties.
1.2.2	2010/10/28	Changed	Added the --omit-nokey-all option.
1.2.1	2010/07/29	Changed	Changed the document format.
1.2.0	2009/12/11	Changed	• Updated the binary file version to 2.1.0.0. • Added support for the L4, A4, and ETC1A4 texture formats. • Removed the `--banner` option.
1.1.0	2009/11/13	Changed	• Changed the (bclim) texture image format. Additional information was moved to the end of the file because of texture alignment requests.
1.0.0	2009/10/30	-	• Initial version.