WiiWare Technical Overview

Justin Braach

Senior Software Engineer

Software Development Support Group

Wii SUMMIT 2008

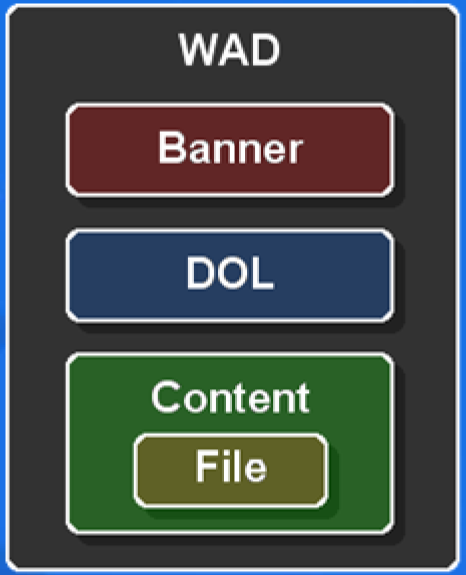
WiiWare vs. Wii discs

- Can do (almost) anything a disc game can do
  - Controllers and Peripherals
  - MP Communication with the Nintendo DS
  - Nintendo WFC
  - WiiConnect24
  - Add-on Content (Data Titles)

- Only real restriction is size
  - WiiWare is an addition to the Wii SDK

WiiWare Data Hierarchy

- WAD
  - Top level container
- Banner
  - Icon and Banner data
- DOL
  - Game Executable
- Content
  - Categorized as User or Shared data*
  - Contains Files
- File
  - Data



* Each item is described in detail in the following slides.

WiiWare Size Restrictions

- WAD Sizes (16 or 40MB)
  - Shared content does not count against size
  - Banner, DOL, and User Content does

WiiWare Size Restrictions

- Save Data
  - Same as disc title (16MB max)
  - Can be shared between NAND applications
    - Episodic content
    - Sequels

NADK Overview

- Components for WiiWare development
  - CNT Library for NAND access
  - makeWad tool for WAD creation
  - Nmenu tool for WAD testing
  - Wii Bitmap Font (Shared Data)
  - Wrist Strap screens (Shared Data)

NADK: CNT Library

- Library features
  - – DVD mode for rapid prototyping/development
  - – NAND mode for final development and testing
  - – Easy to switch back and forth between modes
  - – Cannot make disc games using CNT

CNT Data Hierarchy

- WAD files can contain several Contents
  - Indices start at 2 for WiiWare
  - Indices must be sequential
  - Limit of 8 User Contents
  - No limit to number of Shared Contents

- A Content contains one or more files

CNT File Access Process

- Identical calls in DVD and NAND modes
  - Uses DVD functions by default
  - Define **NANDAPP** to use NAND functions
  - See cntdemo for a usage example

CNT File Access Process

- Basic order of operations
  - – Initialize a Content Handle
  - – Open the file
  - – Manipulate the file
  - – Close the file
  - – Release the Content Handle

CNT File Access Process

- Basic order of operations
  - Initialize a Content Handle

```
CNTInitHandle( ID, &Handle, ... );
```

  - Open the file
  - Manipulate the file
  - Close the file
  - Release the Content Handle

CNT File Access Process

- Basic order of operations
  - Initialize a Content Handle
  - Open the file

```
CNTOpen( &Handle, Name, &FileInfo );
CNTFastOpen( &Handle, id, &FileInfo );
```

  - Manipulate the file
  - Close the file
  - Release the Content Handle

CNT File Access Process

- Basic order of operations
  - Initialize a Content Handle
  - Open the file
  - Manipulate the file

```
CNTRead( &FileInfo, ... );
```

  - Close the file
  - Release the Content Handle

CNT File Access Process

- Basic order of operations
  - Initialize a Content Handle
  - Open the file
  - Manipulate the file
  - Close the file

    ```
    CNTClose( &FileInfo );
    ```
  - Release the Content Handle

CNT File Access Process

- Basic order of operations
  - Initialize a Content Handle
  - Open the file
  - Manipulate the file
  - Close the file
  - Release the Content Handle

```
CNTReleaseHandle( &Handle );
```

CNT: DVD mode

```
dvddata/
    content2/
        subdir/
            userfile1.dat
        userfile2.dat
        ...
    content3/
        wbf1.brfna
        wbf2.brfna
```

```
MyGame.wad
    content2.arc
        subdir/
            userfile1.dat
        userfile2.dat
        ...
    WiiBitmapFont.arc
        wbf1.brfna
        wbf2.brfna
```

CNT: DVD mode

```
dvddata/
   content2/
      subdir/
         userfile1.dat
      userfile2.dat
      ...
   content3/
      wbf1.brfna
      wbf2.brfna
```

```
MyGame.wad
   content2.arc
      subdir/
         userfile1.dat
      userfile2.dat
      ...
   WiiBitmapFont.arc
      wbf1.brfna
      wbf2.brfna
```

CNT: DVD mode

```
dvddata/
   content2/
      subdir/
         userfile1.dat
      userfile2.dat
      ...
   content3/
      wbf1.brfna
      wbf2.brfna
```

```
MyGame.wad
   content2.arc
      subdir/
         userfile1.dat
      userfile2.dat
      ...
   WiiBitmapFont.arc
      wbf1.brfna
      wbf2.brfna
```

DVD mode User Content

- Create a directory for each User Content
  - Directory names use the format contentN
  - Where N is the Content ID number

```
dvddata/
   content2/
      subdir/
         userfile1.dat
      userfile2.dat
      ...
```

Organizing User Content

- Subdirectories and Archives are OK
  - All data must be located inside a Content
  - Data can be compressed (will discuss later)

- Remember that size is the basic restriction
  - No limit to numbers of files and directories
  - Each file adds size to the Content FST
  - Use fewer files and directories if possible

DVD mode Shared Content

- Extract each shared arc into a content directory
  - Don't mix Shared Data with User Data!
  - Don't mix Shared Data with other Shared Data!
  - Use darchD Tool

```
$ darchD.exe -x <content dir> <arc name>
```

- Extracted data is for DVD mode only
  - Don't re-archive the extracted data
  - Use original arc files when creating the WAD

CNT: NAND mode

- User Content
  - Create arc files from content directories
  - Run darchD from inside the Content directory

```
$ cd <contentN dir>
$ darchD.exe -c <files/dirs> <arc name>
```

CNT: NAND mode

- Shared Content
  - Nintendo Provided arc files only
  - Wrist Strap Screens (NADK)
  - Wii Bitmap Font (NADK)
  - Home Button Menu data (HBM SDK)

NAND and DVD Differences

- No asynchronous read support in CNT
  - Background loading needs to be threaded
  - No streaming support from NAND

- Other minor differences
  - DVD read return values differ from NAND
  - Read and Seek speed differences

CNT Cautions

- DVD mode is emulating NAND mode
- NAND mode does not emulate DVD mode
- Don't share CNTHandle structures between threads

Before Building the WAD File

- Banner/Icon data
  - Required to run game from Wii System Menu
  - Created with the same tools as disc games
  - Banner does count against WAD size
  - See Icon and Banner Creation Tools package for details

Testing Banner/Icon data

- Do not add untested data to the WAD!
- Check data using Wii Menu Disc Channel
  - Copy opening.bnr file to $DvdRoot directory
  - Run an ELF on the NDEV
  - The icon will be displayed in the Disc Channel
  - Select the Disc Channel to view the banner

Before Building the WAD File

- setnparentalcontrol
  - Required for Master Submissions
  - Script just changes setting
  - makeWad tool burns setting into WAD
  - See NADK man page for argument list

```
$ setnparentalcontrol <Ratings List>
```

Before Building the WAD File

- setcountrycode
  - Used to set Country Code for both disc and NAND applications

```
$ setcountrycode <jp|us|eu>
```

Before Building the WAD File

- makeDol
  - Converts application ELF into DOL format
  - DOL is added to WAD as Content ID 1

```
$ makeDol -d <dol file> -f <elf file>
```

Building the WAD File

- makeWad
  - Man page is located in NADK manual
  - Important Tool Options

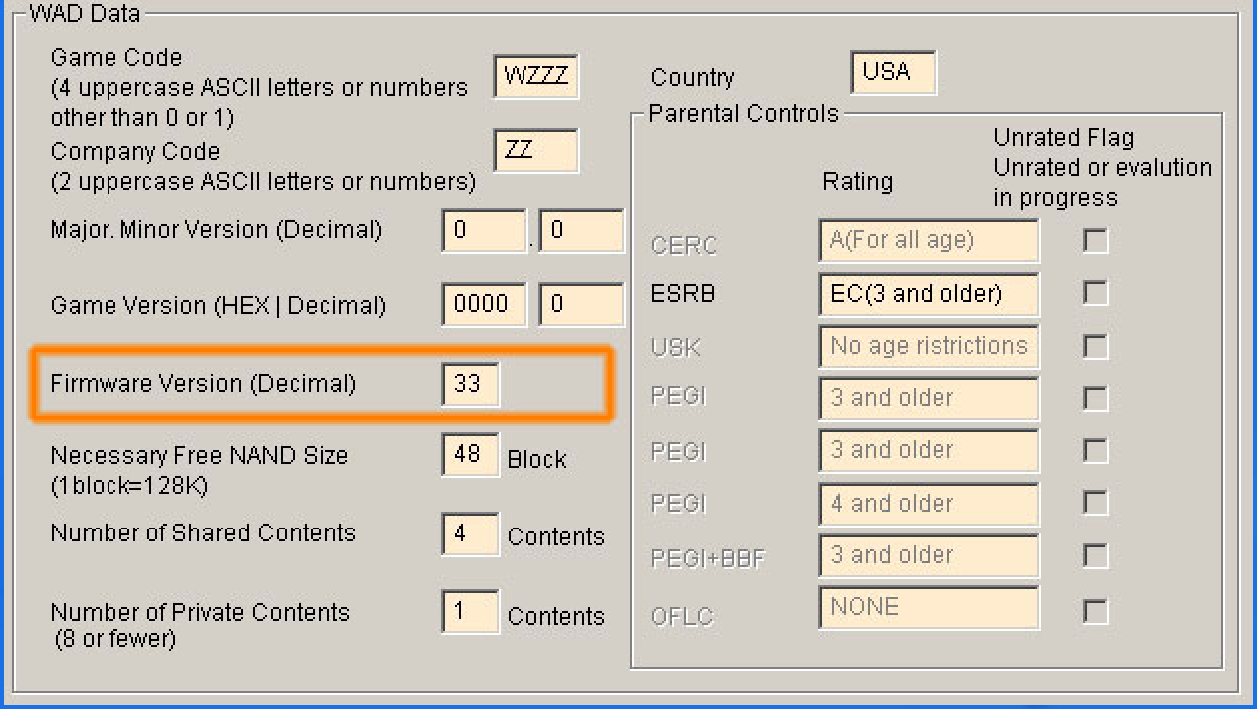
```
-n <titleName>
-m <bannerFile>
-l <DOL,content2,...>
-T <flag1,flag2,...>
```

Revolution Master Editor for WAD

- Information to track during development
  - – Firmware Version (Decimal)
  - – Necessary Free NAND Size
  - – Number of Private Contents

Revolution Master Editor for WAD

- Firmware Version (Decimal)



WAD Data

Game Code
(4 uppercase ASCII letters or numbers other than 0 or 1) WZZZ

Company Code
(2 uppercase ASCII letters or numbers) ZZ

Major. Minor Version (Decimal) 0 . 0

Game Version (HEX | Decimal) 0000 0

Firmware Version (Decimal) 33

Necessary Free NAND Size
(1block=128K) 48 Block

Number of Shared Contents 4 Contents

Number of Private Contents
(8 or fewer) 1 Contents

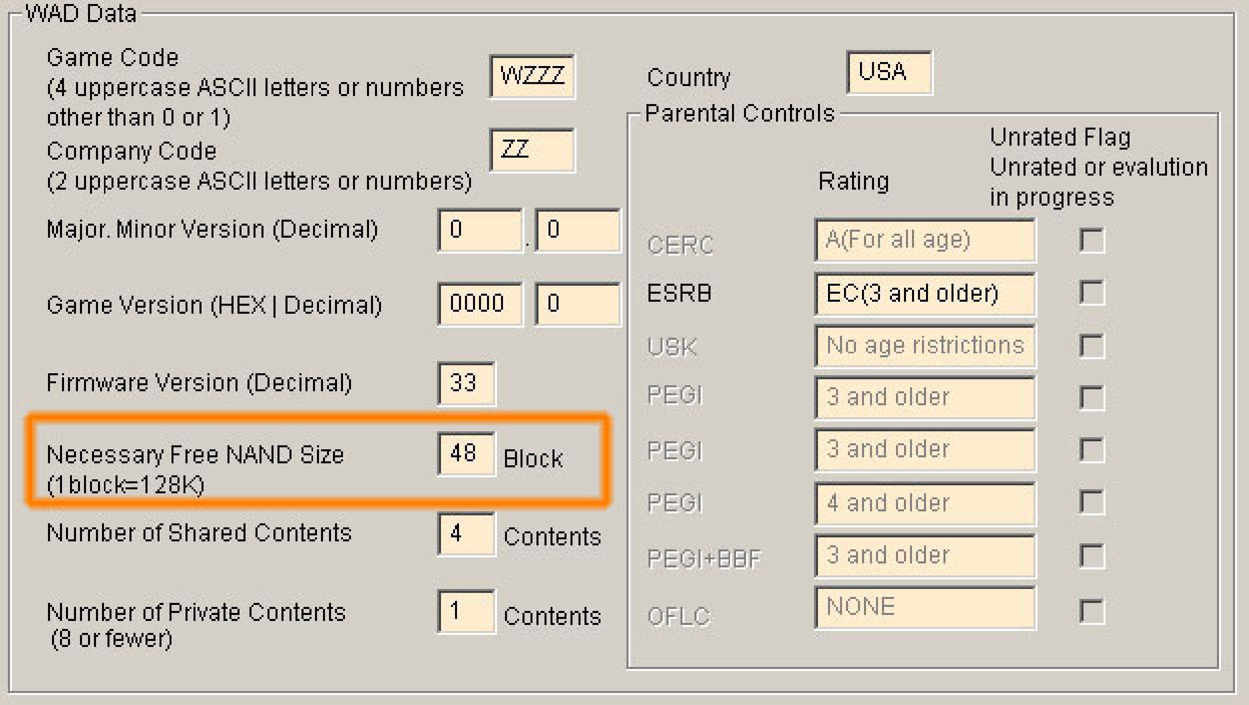
Country USA

Parental Controls

	Rating	Unrated Flag Unrated or evalution in progress
CERC	A(For all age)	□
ESRB	EC(3 and older)	□
USK	No age ristrictions	□
PEGI	3 and older	□
PEGI	3 and older	□
PEGI	4 and older	□
PEGI+BBF	3 and older	□
OFLC	NONE	□

Revolution Master Editor for WAD

- Necessary Free NAND Size



WAD Data

Game Code (4 uppercase ASCII letters or numbers other than 0 or 1): WZZZ

Company Code (2 uppercase ASCII letters or numbers): ZZ

Major. Minor Version (Decimal): 0 . 0

Game Version (HEX | Decimal): 0000 | 0

Firmware Version (Decimal): 33

Necessary Free NAND Size (1block=128K): 48 Block

Number of Shared Contents: 4 Contents

Number of Private Contents (8 or fewer): 1 Contents

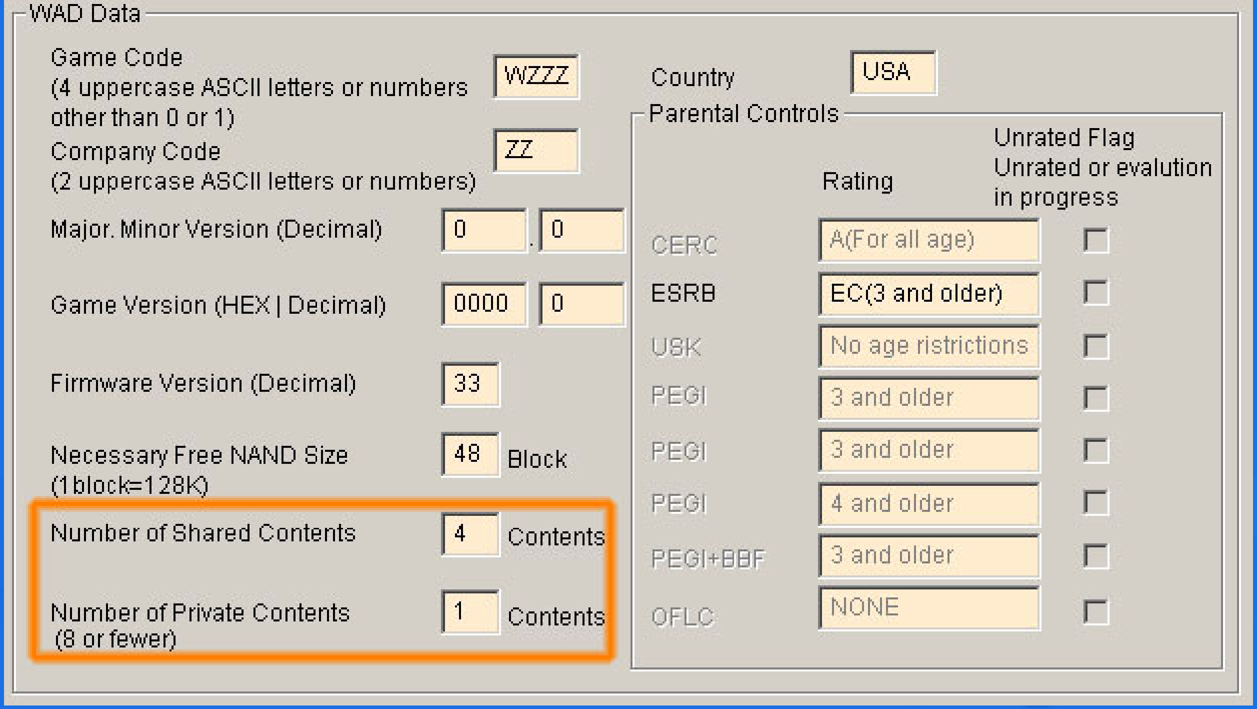
Country: USA

Parental Controls

	Rating	Unrated Flag Unrated or evalution in progress
CERC	A(For all age)	□
ESRB	EC(3 and older)	□
USK	No age ristrictions	□
PEGI	3 and older	□
PEGI	3 and older	□
PEGI	4 and older	□
PEGI+BBF	3 and older	□
OFLC	NONE	□

Revolution Master Editor for WAD

- Number of Private Contents



WAD Data

Game Code (4 uppercase ASCII letters or numbers other than 0 or 1): WZZZ

Company Code (2 uppercase ASCII letters or numbers): ZZ

Major. Minor Version (Decimal): 0 . 0

Game Version (HEX | Decimal): 0000 | 0

Firmware Version (Decimal): 33

Necessary Free NAND Size (1block=128K): 48 Block

Number of Shared Contents: 4 Contents

Number of Private Contents (8 or fewer): 1 Contents

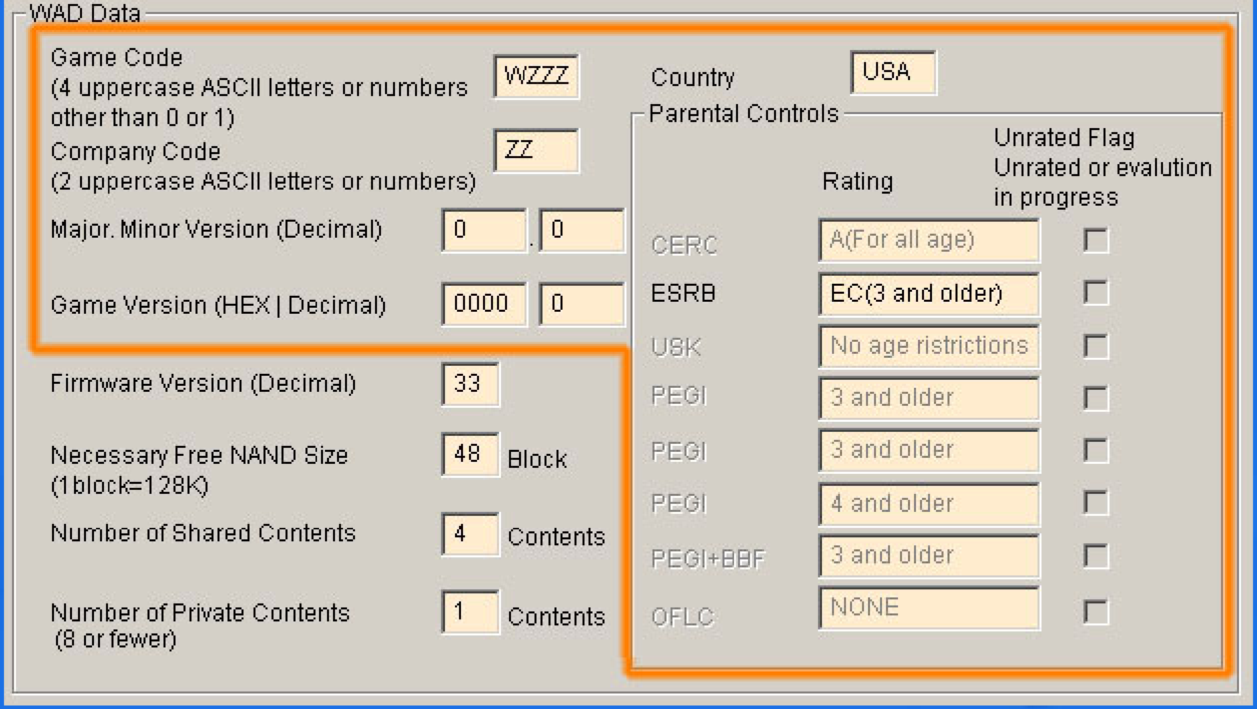
Country: USA

Parental Controls

	Rating	Unrated Flag Unrated or evalution in progress
CERO	A(For all age)	□
ESRB	EC(3 and older)	□
USK	No age ristrictions	□
PEGI	3 and older	□
PEGI	3 and older	□
PEGI	4 and older	□
PEGI+BBF	3 and older	□
OFLC	NONE	□

Revolution Master Editor for WAD Submission Information

- Lotcheck talk will cover in more detail



WAD Data

Game Code (4 uppercase ASCII letters or numbers other than 0 or 1)	WZZZ
Company Code (2 uppercase ASCII letters or numbers)	ZZ
Major. Minor Version (Decimal)	0 . 0
Game Version (HEX | Decimal)	0000 | 0
Firmware Version (Decimal)	33
Necessary Free NAND Size (1block=128K)	48 Block
Number of Shared Contents	4 Contents
Number of Private Contents (8 or fewer)	1 Contents

Country: USA

Parental Controls

	Rating	Unrated Flag Unrated or evalution in progress
CERC	A(For all age)	□
ESRB	EC(3 and older)	□
USK	No age ristrictions	□
PEGI	3 and older	□
PEGI	3 and older	□
PEGI	4 and older	□
PEGI+BBF	3 and older	□
OFLC	NONE	□

Nmenu Tool

- Loads applications into NAND
  - – Located in RVL_SDK/RVL/bin/tools/
  - – Will load WAD files from DVD or SD Card
    - WAD files only
    - $DvdRoot/viewer (DVD)
    - Can navigate directories on SD Card

Nmenu Tool

- Works on both NDEV or RVT-R/H units
  - Does not work on retail hardware
  - Must be mastered and burned to disc for RVT-R use
- Will accept command line arguments
  - See NADK man page for all arguments
  - Auto-Load and/or Auto-Execute WAD by name
  - Note: Auto-load works from DVD only

Development Environment: Nintendo Build Tools & Make

- modulerules
  - NANDAPP (enables WAD rules)
  - CNT_IDX variable (arc generation)
  - Many other options and variables
  - See section marked for NAND application
- buildwad.bat
  - Starts make with preset arguments
  - Edit to fit your needs

Development Environment: Nintendo Build Tools & Make

- wadbuildrun.bat
  - – Convenience batch file for building and running via make
  - – Calls buildwad.bat followed by Nmenu

- wadrun.bat
  - – Convenience batch file loading and running an existing WAD file
  - – Calls Nmenu with preset arguments

Debugging NAND Applications: CodeWarrior

- For Makefile projects, see example in NADK
  - Bottom of manual page "NADK Sample Demos"
  - CodeWarrior calls make, then starts debugger

- For IDE projects, the method is similar
  - Use wadrun.bat instead of wadbuildrun.bat
  - Argument passed to wadrun.bat is build target name

Target Settings -> GCN Target -> File Name

Debugging NAND Applications: CodeWarrior Tips

- Create NAND targets, Keep DVD targets
  - Duplicate existing Debug and Release targets
  - Add "#define NANDAPP" to C/C++ Preprocessor settings for NAND targets

- Create a Post-Build script
  - Set as BatchRunner PostLinker script
  - Convert ELF to DOL
  - Update any modified data files and archives
  - Compress DOL
  - Build WAD file
  - Copy WAD to $DvdRoot/viewer

Data Compression Overview

- Limit or eliminate duplicate data
  - Use String Tables
  - Reuse models and textures

- Use Common Sense practices
  - Only use software and libraries you need
  - Watch out for duplicated functionality in code
  - Package and compress data files together

Data Compression:
CX Library

- Standard component of Revolution SDK
  - – Support for several compression formats
  - – LZ77(ex), RLE, Huffman, Diff. Filter

- Use ntcompress tool to compress data
  - – See man page for command line options
  - – Remember that bundled files yield better compression

Data Compression: CX Library

- Supports whole file decompression
- Supports decompression in blocks
  - Library provides streaming support for LZ77, Huffman, and RLE data.
  - Streaming functions track write position in output buffer.
  - Read compressed data into a small read buffer one block at a time.
  - Call decompression function for each block until all blocks are loaded.

Data Decompression: File-at-Once vs. Block

- File-at-Once Method
  - Pro: Generally faster than Block Method as there is only 1 read and 1 decompress. This is particularly evident with large files.
  - Con: Requires a dynamically sized read buffer to hold entire compressed file prior to decompression.
  - Con: Dynamic nature makes it prone to fragmentation.

- Block Method
  - Pro: Uses fixed amount of memory independent of file size.
  - Pro: No memory fragmentation if the read buffer is static.
  - Con: Can be slower than the File-at-Once method due to multiple file reads. The read buffer size however can be adjusted to suit the data and minimize reads.

Data Compression: DOL

- WiiWare specific
  - Doesn't work with disc applications
  - Cuts DOL size down significantly

- Compress with ntcompress
  - Use LZ77ex compression

```
$ ntcompress –A32 -lex <dol file>
```

Compression Best Practices: Textures

- GX texture formats
  - CMP (a.k.a. S3TC/DXT1)
  - Indexed (I8, IA8, I4, IA4)
  - Use TexConv tool or NintendoWare Photoshop Plugin
- JPEG Library
  - Can produce better visible quality than CMP for certain textures
  - More details during Middleware Talk

Compression Best Practices: Texture Tips

- Split RGBA images into RGB and A
  - Use CMP format for RGB layer, CMP or Index for A
  - Recombine in TEV (uses 2 stages)
- Use color swapping tricks
  - Use vertex color or material to tint greyscale images
  - Use palette swapping for indexed textures
- Size textures according to their use
  - Only store mipmap levels you will use

Compression Best Practices: Geometry

- Use fixed-point data
  - – CPU and GPU Hardware supported
  - – Faster than floating point data
- Use indexed geometry
  - – Great for display lists
  - – Share component arrays between models

Compression Best Practices: Geometry Tips

- Split data into component arrays
  - Positions, Normals, Colors, Etc.
  - Eliminate duplicates in each array

- Combine unique models where possible
  - Geometry that is used together, but only in one place
  - Combined geometry shares one data pool

- Use Instanced geometry
  - Don't create a unique model for each game object
  - This also applies to UI elements

Compression Best Practices: Sound

- Use DSP-ADPCM format for wave data
  - Hardware supported (no cost)
  - Excellent compression (3.5:1)
- Use MIDI (or MOD) for music
  - Significantly smaller than wave data
  - Remember: No support for streaming from NAND
- Compress sound banks inside Content file
  - Decompress to RAM for playback
  - Extra benefit is faster loading times

Compression Best Practices: General

- Cannot compress Content arc files
  - – Can compress the data inside
- Try different compression formats
  - – Better results for different data types

Compression Best Practices: General

- Reuse data where possible
  - Take advantage of instancing and texture tricks
  - Take advantage of Shared Content
- Package and compress data by usage
  - Data used throughout the game
  - Data used only in a single area/level
  - Front-end data

Home Button Menu: WiiWare version

- 3 Button version
  - – Link against homeButton.nwm.a

- Contains Operations Guide screen
  - – Developer created – not in shared content
  - – Contains simple game instructions only
  - – Minimum of English content only

Home Button Menu: “Game to Manual to Game”

- Saving game state before exit
  - This is a design decision – not required
  - Can save entire game state, or just current level
  - OK to save data to use save game area

- OSLaunchManualViewer
  - One 512 byte length argument
  - Must be NULL or a NULL terminated string
  - Can encode game state information into string

Home Button Menu: “Game to Manual to Game”

- Dummyviewer.wad
  - Comes in NADK package (RVL_SDK/dvddata/viewer)
  - Use Nmenu to load into NAND
  - Only for testing game flow, does not display manual
- Return to game
  - Manual Viewer application launches your application
  - Argument passed to OSLaunchManualViewer is passed back
  - If no argument exists assume game was booted directly from menu

WiiWare Online Manual

- Separate from game WAD
  - Not counted against WAD size
  - Submitted as a Zip archive with WAD
- Viewed from Wii Shop Channel
  - Can be viewed before game purchase
  - Game manual information only
  - No advertisements

WiiWare Online Manual Creation

- WiiWare Online Manual Guideline Package
  - Online Manual contents
  - Operations Guide image guidelines
  - See Lotcheck WiiWare talk for more specifics

- WWManTool
  - Tool for viewing Online Manual content on NDEV
  - Separate package on WarioWorld

Questions?

- Contact support@noa.com
- Thank you for listening