

N I N T E N D O

NITRO-SampleTools

3D Sample Converter

(This version supports NITRO-SDK 2.0 PR3 Plus)

Version 1.1.2

**The contents of this document are strictly confidential
and the document should be handled accordingly.**

Contents

1	Introduction	5
2	The Contents of This Package	6
2.1	The Folder Structure for This Package	6
2.1.1	3DSampleConverter.pdf	6
2.1.2	SimplePreview (A Simple Display Program).....	6
2.1.3	obj2nitro (Converter for obj Files)	7
2.1.4	imd2nitro (Converter for imd Files)	7
2.2	Other Information.....	7
3	Simple Display Program	8
3.1	The Simple Display Program	8
3.2	Running the Simple Display Program on the TEG-01 Board	8
3.2.1	Required Execution Environment	8
3.2.2	Verifying Operation	8
3.3	Display Description	9
3.4	How to Operate	9
3.5	How to Display Different 3D Model Data	10
3.5.1	How to Use obj2nitro to Display the Contents of obj Files	10
3.5.2	How to Use imd2nitro to Display the Contents of imd Files	11
3.5.3	Flowchart for Display of 3D Model.....	12
3.5.4	Main Features Supported by the Simple Display Program	13
3.6	Configuration for Display on TEG-01 Board	14
3.7	Other Information.....	14
4	obj2nitro	15
4.1	Function.....	15
4.2	How to Execute	15
4.3	Customizing obj2nitro	16
4.4	Other Information.....	16
5	imd2nitro	17
5.1	Function.....	17
5.2	How to Execute	17
5.3	Customizing imd2nitro	18
5.4	Other Information.....	18
6	Xerces	19

Tables

Table 3-1 How to operate the simple display program	9
Table 3-2 Main features supported by the simple display program	13

Figures

Figure 3-1 Image of the simple display program's display screen	9
Figure 3-2 3D Model Display Flowchart	12

Revision History

Version	Revision Date	Description of revisions
1.1.2	08/02/2004	<ul style="list-style-type: none">- Confirmed operation with NITRO-SDK 2.0 PR3 Plus.- Added 3.6 "Configuration for Display on TEG 01 Board."- Revised simple display program.- Name of ROM image file changed from main.bin to main.srl.- Updated model.h (added #pragma warn padding).
1.1.1	06/30/2004	<ul style="list-style-type: none">- Verified operation with NITRO-SDK 1.2.- Revised simple display program.- Deleted unnecessary G3 command from the CalcModelMatrix() function.
1.1.0	04/26/2004	<ul style="list-style-type: none">- Support for NITRO-SDK 1.0.- Support for A3I5 translucent texture.- Revised the name of the intermediate fille parser from MiddleFmt to IntermediateFileParser.- Revised "Figure 3-2 TEG-01 Board Display Flowchart".- Added to "3.5.4 Main Features Supported by the Simple Display Program".- Added caution to "3.6 Other Information".- Revised obj2nitro<ul style="list-style-type: none">* Fixed bug that prevented Obj files without normal vectors or group information from being converted properly.- Revised imd2nitro<ul style="list-style-type: none">* Fixed bug that prevented proper recognition when only imd file names were input.- Revised simple display program<ul style="list-style-type: none">* Revised VRAM configuration.* Fixed bug preventing translucent textures from being displayed as translucent.
1.0.0	01/30/2004	Release

1 Introduction

3DSampleConverter is a suite of sample programs for converting the 3D model data files exported from 3D CG tool (the Wavefront OBJ files or imd files) into C source text files and then displaying them in a simple way with IS-NITRO-EMULATOR (hardware).

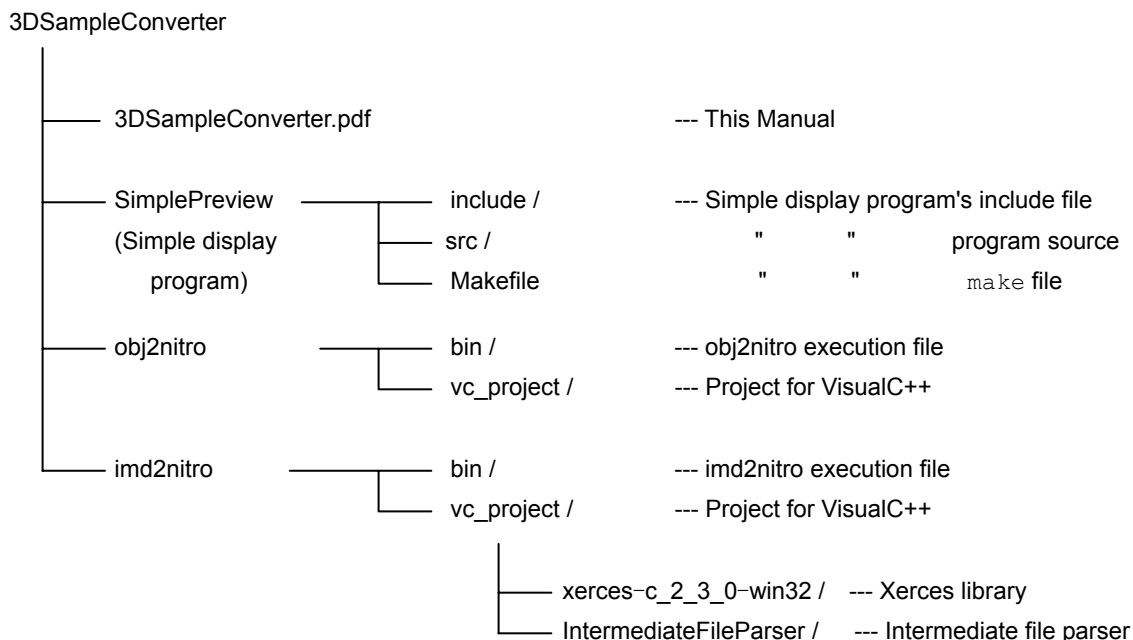
This package contains the following three programs:

- Simple display program: SimplePreview
- obj file converter: obj2nitro
- imd file converter: imd2nitro

2 The Contents of This Package

2.1 The Folder Structure for This Package

The contents of this package are stored using the following folder structure:



The programs in this package all contain source files.

Because these are sample programs, we cannot accommodate any requests to change the programs.

2.1.1 3DSampleConverter.pdf

This manual is stored as a PDF file. The manual explains how to set up and use 3DSampleConverter.

2.1.2 SimplePreview (A Simple Display Program)

The simple display program located in the SimplePreview folder is a sample program that uses the NITRO-SDK to display 3D models on NITRO.

2.1.3 obj2nitro (Converter for obj Files)

This is a converter that takes the obj file exported from the 3D CG tool (the Wavefront OBJ file, with the extension .obj) and converts it into a C source format file. It is located in the obj2nitro folder.

2.1.4 imd2nitro (Converter for imd Files)

This is a converter that takes the imd file exported from the 3D CG tool (the NITRO Intermediate file's model data file, with the extension .imd) and converts it into a C source format file. It is located in the imd2nitro folder.

The "IntermediateFileParser" folder in the imd2nitro folder is a sub-project for reading intermediate files.

Under the xerces-c_2_3_0-win32 folder is the Xerces library, which is used by imd2nitro to read XML files.

2.2 Other Information

The operation of this package has been verified with NITRO-SDK 2.0 PR3 Plus.

It will not operate correctly on any versions earlier or later than this.

3 Simple Display Program

3.1 The Simple Display Program

This is a sample program for the simple display of 3D models on NITRO.

Use one of the file converters (`obj2nitro.exe` or `imd2nitro.exe`) to create a `model.c` file, then substitute this for the same-named file in the simple display program and execute `make` on the file. This creates a NITRO ROM image file. You can display this file with the IS-NITRO-EMULATOR.

3.2 Running the Simple Display Program on the TEG-01 Board

3.2.1 Required Execution Environment

Use of the simple display program requires configuration of the NITRO-SDK, CodeWarrior for NITRO, and Cygwin.

3.2.2 Verifying Operation

(1) Verify that the NITRO development environment has been set up correctly.

The development environment has been set up correctly if IS-NITRO-EMULATOR can be used to display a sample normally after building the sample program using `make`.

(2) Copy the `3DSampleConverter` folder to somewhere on the PC (anywhere).

(3) Start Cygwin and move the `3DSampleConverter/SimplePreview` folder copied in step 2.

(4) Execute the `make` command.

(5) Load the ROM Image file (`main.srl`) in the current folder into IS-NITRO-EMULATOR.

(6) The system is operating normally if the picture shown in Fig. 3-1 is displayed.

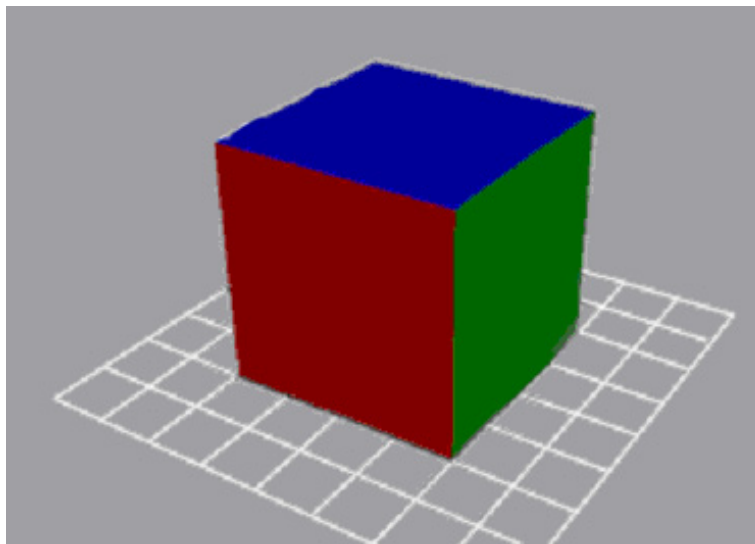


Figure 3-1 Image of the simple display program's display screen

3.3 Display Description

The program displays a wireframe grid rendered in the XZ plane and 3D models.

3.4 How to Operate

The simple display program is controlled using the following button operations.

Table 3-1 How to operate the simple display program

	Explanation
+Control Pad	Moves the position of the eye point.
SELECT	Toggles between show/hide the wireframe grid.
START	Changes the background color.
A / B Buttons	Changes the distance between the eye point and the attention point (the look-at point).
L / R Buttons	Changes the height of the attention point (the sight point).

3.5 How to Display Different 3D Model Data

In the source codes of the simple display program, all of the 3D model data is stored in the `model.c` file. You can display different 3D model data on the IS-NITRO-EMULATOR by following the procedure outlined below to replace the `model.c` file and re-execute `make`.

3.5.1 How to Use `obj2nitro` to Display the Contents of `obj` Files

(1) Export the `obj` file from the 3D CG tool.

(2) Start Cygwin and go to the `3DSampleConverter/SimplePreview` folder.

(3) Enter the following command in Cygwin to convert the `obj` file into a `model.c` file.

```
.. /obj2nitro/bin/obj2nitro.exe [specify obj file]
```

(Alternately, you can convert the `obj` file by dragging & dropping it on the `obj2nitro.exe` icon)

(4) Upon conversion, a `model.c` file is created in the same folder as the `obj` file. Take this `model.c` file and move it to the `SimplePreview/src` folder using this command:

```
mv [specify the model.c file that converted from the obj file] ./src
```

(5) Execute "`make`."

(6) The ROM Image file (`main.srl`) in the current folder is loaded into the IS-NITRO-EMULATOR.

3.5.2 How to Use imd2nitro to Display the Contents of imd Files

You can display the 3D models described in imd files by loading them into the IS-NITRO-EMULATOR following the procedure outlined below, to replace the `model.c` file in the simple display program.

(1) Export the imd file from the 3D CG tool using the NITRO Intermediate File Plug-in.

(To learn how to configure each NITRO Intermediate File Plug-in, read the manual that explains how to install and use that particular NITRO-System plug-in.)

(2) Start Cygwin and move to the `3DSampleConverter/SimplePreview` folder.

(3) Enter the following command in Cygwin to convert the imd file into a `model.c` file.

```
../imd2nitro/bin/imd2nitro.exe [specify the imd file]
```

(Alternately, you can convert the imd file by dragging & dropping it on the imd2nitro.exe icon)

(4) Upon conversion, a `model.c` file is created in the same folder as the imd file. Move the `model.c` file to the `SimplePreview/src` folder using this command:

```
mv [specify the model.c file that converted from the imd file] ./src
```

(5) Execute "make."

(6) The ROM Image file (`main.srl`) in the current folder is loaded into the IS-NITRO-EMULATOR.

3.5.3 Flowchart for Display of 3D Model

This flowchart shows the steps involved in using obj2nitro and imd2nitro to display the contents of obj files and imd files.

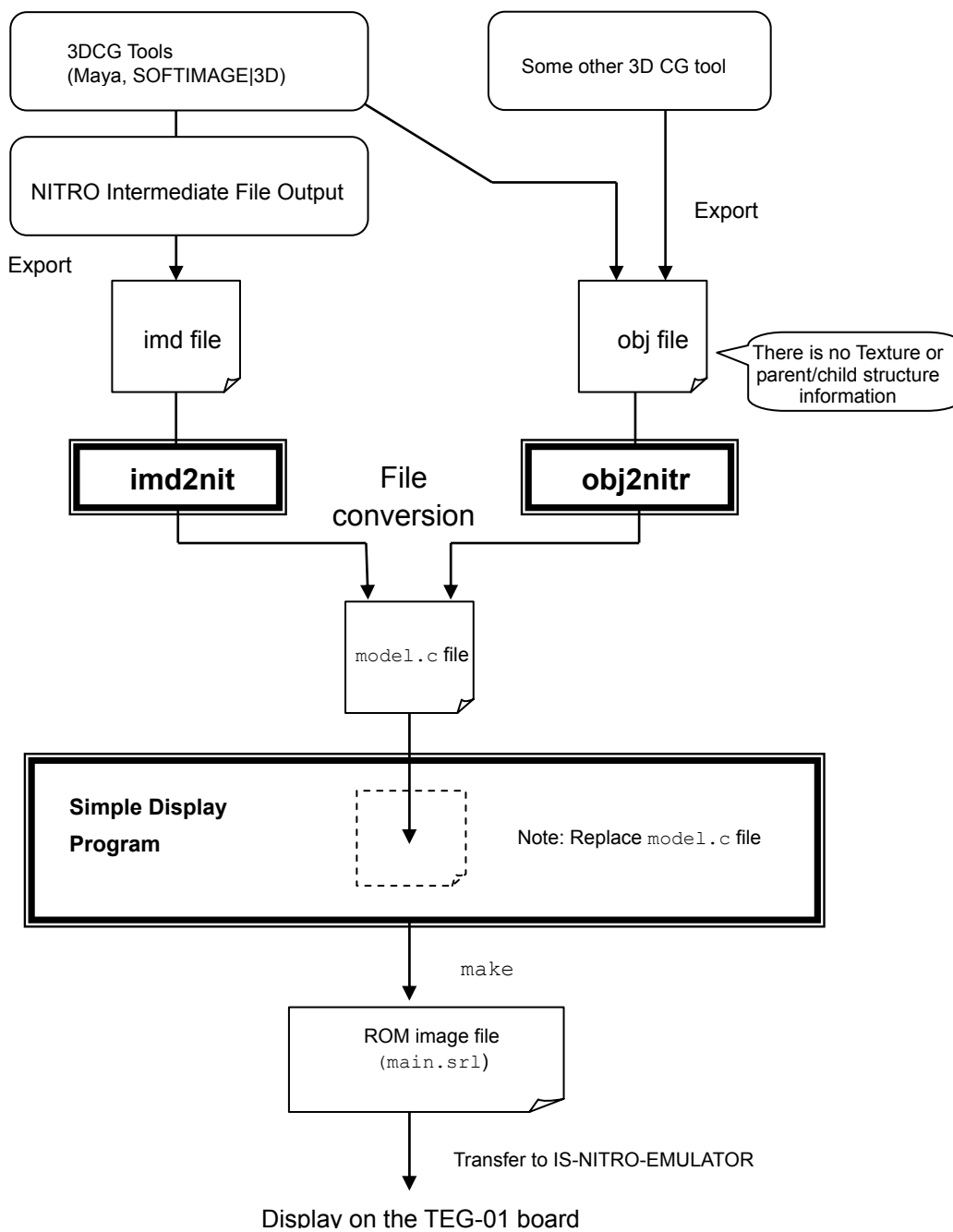


Figure 3-2 3D Model Display Flowchart

3.5.4 Main Features Supported by the Simple Display Program

The simple display program supports the following types of content in obj files and imd files.

Note: The `model.c` file created using `obj2nitro` or `imd2nitro` only stores information for display with the simple display program

Table 3-2 Main features supported by the simple display program

Item	obj2nitro	imd2nitro
Polygon shapes (vertex position coordinates)	○	○
Shading (normal vectors)	○	○
Display vertex colors	—	○
Display textures	×	○
Parent/child structure of nodes	—	○
Node visibility (show / hide)	—	○
Material color (diffuse, ambient, specular)	○	○
Display polygon alpha and alpha blending	○	○
Polygon display faces (front face, back face, both faces)	— (Always display both faces)	○
Display wireframes	—	○
Light 0 ON/OFF	— (Always ON)	○
Texture displays in the formats below: (4-, 16-, and 256-color palettes, 4x4 texel compressed, A5I3 Translucent, A3I5 Translucent, Direct Texture).	—	○
Display texture repeat / flip	—	○
Enable palette's color 0 setting. (Use color 0 or not)	—	○
Full weight envelope. (But number of matrices needed for drawing is 31 or less.)	—	○
Maya's SegmentScaleCompensate — ON	—	×
Off display for SOFTIMAGE 3D classic scaling	—	×
Display billboard	—	×
Material color (emission)	—	×
Switch polygon mode (modulation, decal, other)	—	×
Texture matrix manipulation	—	×
Display toon / highlight	—	×
Weight envelope (with weight set for multiple nodes)	—	×
Fog	—	—
Edge marking	—	—
Show various animations	—	—

○ = Supported

× = Not supported

— = File is not supported

3.6 Configuration for Display on TEG-01 Board

The default settings of the simple display program's makefile are for display on the IS-NITRO-EMULATOR (hardware). To display on the TEG-01 board, change the following lines in the makefile and execute make.

Change

```
NITRO_PLATFORM = TS
```

To

```
NITRO_PLATFORM = TEG
```

3.7 Other Information

- With the simple display program, matrix calculations are performed by the CPU in order to display full-weight envelope models.
All matrices needed for drawing the model are calculated and assigned to a 31-element matrix stack, after which Geometry commands are sent in order to draw the polygons. If the model requires more than 31 matrices for drawing it cannot be displayed.
- 3D models are displayed after undergoing an overall scaling, in which XYZ are scaled to fit in a global coordinates grid of -8.0 to +8.0.
- The model cannot be displayed if the name of a node or texture begins with a number.
The above naming settings are not a problem for NITRO Intermediate files but will result in an error on the simple display program because it treats texture and node names in imd files as names of C arrays. If characters other than alphanumeric are used in the name character string, those characters will automatically be converted to underscores ("_").
- A3I5 translucent texture cannot be displayed on the TEG-01 Board.

4 obj2nitro

4.1 Function

This converts a Wavefront OBJ file (with the extension `.obj`) into a C source format file named `model.c`.

By substituting this newly created `model.c` file for the one already in the simple display program and executing `make` you can use the IS-NITRO-EMULATOR to display the 3D model data contained in the `obj` file.

4.2 How to Execute

Use the following syntax to execute `obj2nitro.exe`, which is stored in the `3DSampleConverter/obj2nitro/bin` folder.

`obj2nitro.exe [options] input-file.obj`

Options: `-h / -help` : Show help
 `-v / -verbose` : Show process details
 `-q / -quiet` : Do not show

Note: The `model.c` file that results from the conversion is output to the same folder where the `obj` file resides.

You can also convert the `obj` file by dragging & dropping it on the `obj2nitro.exe` icon

4.3 Customizing obj2nitro

This package contains the source program for obj2nitro. You can customize the obj2nitro source if you want to convert the obj file into your own file format.

obj2nitro is created in the following environment:

- Windows2000 sp4
- VisualStudio.net version2003

Procedure for customizing obj2nitro:

- (1) Use VisualStudio.net to open the `obj2nitro.sln` file, which is stored in the `3DSampleConverter/obj2nitro/vc_project` folder.
- (2) Customize the source file as needed and build the project.

4.4 Other Information

- Only polygon objects are supported. Objects that use curved lines and curved surfaces are not supported.
- To load an mtl file (material settings file) at the same time as an obj file, the two files need to be located in the same folder. A white material will be set if an mtl file is not described inside the obj file or if the mtl file described inside the obj file cannot be located.
- Command packing is not performed on the NITRO Geometry commands exported to the `model.c` file.
- Textures are not yet supported. Texture coordinates are not exported to the `model.c` file even if they exist inside the obj file.
- Polygons are all exported as either triangular polygons or quadrilateral polygons. obj2nitro does not internally conduct an optimization process on connected triangular polygons or connected quadrilateral polygons. Any polygons of five or more sides are divided into triangular shapes for export.

5 imd2nitro

5.1 Function

This converts an imd file (with the extension `.imd`) into a C source format file named `model.c`.

By substituting this newly created `model.c` file for the one in the simple display program and executing `make` you can use the IS-NITRO-EMULATOR to display the 3D model data contained in the imd file.

5.2 How to Execute

Use the following syntax to execute `imd2nitro.exe`, which is stored in the `3DSampleConverter/imd2nitro/bin` folder.

imd2nitro.exe [options] input-file.imd

Options: `-h / -help` : Show help
 `-v / -verbose` : Show process details
 `-q / -quiet` : Do not show

Note: The `model.c` file that results from the conversion is output to the same folder where the imd file resides.

You can also convert the imd file by dragging & dropping it on the `imd2nitro.exe` icon.

In order to execute `imd2nitro.exe` you need `xerces-c_2_3_0.dll`

5.3 Customizing imd2nitro

This package contains the source program for imd2nitro.

You can customize the imd2nitro source if you want to convert the imd file into a file that uses your own file format.

The imd2nitro program maintains information in a class that has the same data structure as the imd elements. You can export data to a file in your own data format by consulting the `imd2nitro.cpp` file and rewriting the export portion.

The imd2nitro program uses the Xerces library internally to read imd files in XML format.

imd2nitro is created in the following environment:

- Windows2000 sp4
- VisualStudio.net version2003
- Xerces C++ v2.3.0 (included in this package)

Procedure for customizing imd2nitro:

- (1) Use VisualStudio.net to open the `imd2nitro.sln` file, which is stored in the `3DSampleConverter/imd2nitro/vc_project` folder.
- (2) Customize the source file as needed and build the project.

5.4 Other Information

- Polygon vertices have already been optimized by the time of the imd file conversion. imd2nitro accesses information inside the imd file to export the vertex-related Geometry commands used for triangular polygons, quadrilateral polygons, connected triangular polygons and connected quadrangular polygons. Command packing is not performed.
- An error will occur if imd2nitro attempts to convert the following types of imd files:
 - A model requiring more than 31 matrices for drawing
 - A model that uses a weight envelope
 - A model that has more polygons or vertices than NITRO permits

6 Xerces

The package incorporates software from Apache Software Foundation (<http://www.apache.org/>)

The Apache Software License, Version 1.1

Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
"This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., <http://www.ibm.com>.
For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Windows and Visual Studio are the trademarks or copyrighted trademarks of the Microsoft Corporation.

Metrowerks and CodeWarrior are the trademarks or copyrighted trademarks of Metrowerks, Inc.

Maya is the trademark or copyrighted trademark of Alias Systems Co.

Softimage, SOFTIMAGE|3D, and SOFTIMAGE|XSI are the trademarks or copyrighted trademarks of Avid Technology, Inc.

3ds max is the registered trademark or trademark of Autodesk, Inc./Autodesk Canada, Inc. in the USA and other nations.

All other company or product names are the trademark or copyrighted trademark of the respective company.

© 2003, 2004 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.