MB86290 Series Model Converter V02 for LightWave3D User Manual

Version 1.1

FUJITSU

1. The specifications in this manual are subject to change without notice. Contact our Sales Department before purchasing the product described in this manual.
2. Information and circuit diagrams in this manual are only examples of device applications, they are not intended to be used in actual equipment. Also, Fujitsu accepts no responsibility for infringement of patents or other rights owned by third parties caused by use of the information and circuit diagrams.
3. The contents of this manual must not be reprinted or duplicated without permission of Fujitsu.
4. If the products described in this manual fall within the goods or technologies regulated by the Foreign Exchange and Foreign Trade Law, permission must be obtained before exporting the goods or technologies.

Updated History

Data	Version	Updated part	Updated Contents
2003/1/27	1.0	—	First edition
2003/2/28	1.1	5.2	・Add explanation：Table5.2-k The following sentence is added for restrictions of the corn angle. “It does not correspond to the dynamic corn angle configuration by the envelope.” ・Error correction：It changes the title of Table5.2-1 to “Camera motion function”. ・Error correction：It changes the corresponding condition of the camera zoom to “○”. ・Add explanation：The following sentence is added for the restrictions of the camera zoom. “It does not correspond to the dynamic camera zoom by the envelope”
		10	・Error correction：The following warning message is corrected. Fault：IDS_WARNING_BGIMAGE_NUM "Background image can use only one in a scene." Right：Background image can use only one in a scene. ・Error correction：The following warning message is corrected. Fault：IDS_WARNING_LIGHT_NUM "The number of the light over 8." Correct：The number of the light over 8.

（Page of updated history：1／1）

Preface

■Purpose of the manual

In this manual, the usage of *MB86290 Series Model Converter V02 for LightWave3D* (model converter) is described.

The model converter is a software, which converts the scene file created by the modeling tool, NewTek LightWave3D, to the data format of the scene file to play back by *MB86290 Series 3D Graphics Library V02* (3DGL).

Refer to the following manuals for *MB86290 Series 3D Graphics Library*.

- 『MB86290 Series 3D Graphics Library V02 user manual –The core API』
- 『MB86290 Series 3D Graphics Library V02 user manual –The scene drawing API』

In this manual, as it uses the technical terms of the 3D computer graphics used by LightWave3D, it does not explain about those. If it is necessary, refer to the manual attached to LightWave3D or the manual on LigtWave3D available in the market.

■A registered trademark/ a trademark

LightWave and LightWave 3D are the registered trademark of NewTek, Inc in the United States and other countries.

Micorsoft, Windows, and Windows NT are the registered trademark of Microsoft Corporation in the United States and other countries.

Other described company name and product name are a registered trademark or a trademark for each company.

Outline

1. GENERAL

MB86290 Series Model Converter V02 for LightWave3D (model converter) reads the scene file, the object file, and the image data, which are created by the modeling tool, NewTek LightWave3D(Ver.5.x, Ver.6.x), and transfers the data for MB86290 series graphics controller. This data is called the model data.

The model data is input in the scene drawing API (3DGL scene drawing API) of *MB86290 Series 3D Graphics Library*, and could be played back by MB86290 series graphics controller.

2. FUNCTION

The functions of the model converter are described in the following.

- It transfers not only the object but also the information on the scene of a light, a camera, the motion information of movement/ rotation/ scaling.
- It reads multi scene files, and can put together into one file and output it.
- It can transfer the object file itself or the image data itself.
- If there are same object in the scene file, those object data are managed as one data.
- If there are same image data in the scene file, those image data are managed as one data.
- Except the model data, it creates the model data index file to refer the model data information. The model index file has the index of the various information of inside the model data, and this index is used in 3DGL scene drawing API.

3. OPERATING CONDITIONS

The operation conditions of the model converter are following.

- Microsoft Windows95, Windows98, WindowsNT4.0, or Windows2000 Japanese version and English version
- Over 128MB(Mbyte) of RAM(random access memory)
- Over 16MB of the empty hard disk

4. INPUT/OUTPUT

It shows the input/output files of the model converter.

Table4.1-a Input file

File format	Content	Comment
.lws	LightWave 3D scene file	-
.lwo	LightWave 3D object file	-
.bmp	Microsoft Windows Bitmap	Texture image data
.iff	Interchange File Format	Texture image data
.tga	Truevision Graphics Adapter	Texture image data
.mdx	Extension model data file (*1)	-

*1 Model data format, which could be input and reconfigured.

Table4.1-b Output file

File format	Content
.mdf	Model data file
.mdx	Extension model data file (*1)
.h	Model data index file
.log	Convert log file(*2)
.dat	C language source format model data file
.txt	Text format model data file

*1 Model data format, which could be input and reconfigured.

*2 Output message log of the model converter

5. LightWave 3D CORRESPONDING

It shows the corresponding condition of the model converter on the function of LightWave3D (Ver.5.x Ver.6.x). For the functions, which are not responded, even if it is configured, it ignores in transferring. For some of non-responding functions, it writes the warning message on the log. As non-responding function is configured, the transfer process is not terminated. However, it is not guarantee to transfer the model data, which can be played back. Do not use for the non-responding functions.

5.1. Modeler function

The corresponding condition for the modeler function is following Table5.1.

Table5.1 Corresponding condition for the modeler function

Process function	Corresponding condition
Color editing	◎ ※1
Image editing	◎ ※1
Object data created by spline curve	× ※2
Object data created by MetaNURBS	× ※2
Various plug-in	×

◎：Correspond, ○：Partially correspond×：non-correspond

※１：It correspond only LightWave 3D Ver.6.x.

※２：The object created by the spline curve and MetaNURBS can be transformed, consequently that it is transformed to the polygon and inputs the converter.

5.2. Layout function

The layout functions are categorized in following thirteen functions. The corresponding condition of each function is shown in Tabel5.2-a~Tabel5.2-m.

• Scene(lender option)function	(Table 5.2-a)
• Object function	(Table 5.2-b)
• Surface function	(Table 5.2-c)
• Texture function	(Table 5.2-d)
• Image function	(Table 5.2-e)
• Light function	(Table 5.2-f)
• Camera function	(Table 5.2-g)
• Effect(special effects)function	(Table 5.2-h)
• Record function	(Table 5.2-i)
• Object motion function	(Table 5.2-j)
• Light motion function	(Table 5.2-k)
• Camera motion function	(Table 5.2-l)
• Motion option function	(Table 5.2-m)

Table 5.2-a Scene(lender option)function

Process function	Corresponding condition
Starting frame(FirstFrame)	○ ※1
Ending frame(LastFrame)	○ ※1
Frame step(FrameStep)	○ ※2
Frame／Second(FramePerSecond)	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：It corresponds only the starting/ending frame of the positive value. Also, the starting frame and ending frame must be the starting frame ≦ the ending frame.

※2：It corresponds only the frame step of the positive value.

Table5.2-b Object function

Process function	Corresponding condition
Bone, skelton(Object Skeleton)	×
Displacement Map	×
Metamorph	○ ※1
Object Dissolve	○ ※2
Clip Map	○ ※3
Distance Dissolve	×
Particle/Line Size	×
Polygon Edge	×
Edge of the Cel-Image Condition(Cel-Look Edge)	×
Self Shadow	×
Cast Shadow	×
Receive Shadow	×
Unseen by Rays	×
Unaffected by Fog	×
Null Object	○ ※4
Specify the parent object	×
Inverse Kinematics	×
Various plug-in	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：The configuration of the object color・texture・image for the object of the morphing destination must be same as the configuration of the original object. In LightWave3D, it is possible to configure separately, it is not corresponded in the model converter. If the configured value is differing, it happens to perform normally the transformation.

※2：The object dissolve is used to mask the object of the morphing destination. In the model converter, it can configure only the visible (0%) or the invisible (100%).

※3：Specify the original image for specifying the clip map. As it can specify only the clip map, it is not corresponded in the model converter.

The image size and the wrap configuration to configure in the clip map must be same as the configuration of the original data image.

Specify two value patterns (black and white) for the clip map. If it specifies the image, except the original image, the image does not be normally transformed.

※4：It can specify only the motion information for Null object. It can not configure the parent-child relationship and so on.

Table5.2-c Surface function

Process function	Corresponding condition
Surface Color	○ ※1
Luminosity	○ ※1
Diffuse Level	○ ※1
Specular Level	○ ※1
Glossiness	○ ※1
Reflectivity	×
Transparency	◎
Inflection index	×
Semi-transparency	×
Smoothing	○ ※2
Double Sided	×
Bump Map	×
Alpha Channel	×
Shadow Alpha	×
Special Buffers	×
Glow Effect	×
Outline Lender(Outline Only)	×
Color Highlights	×
Color Filter	×
Additive of the transparency(Additive)	×
Sharp Terminator	×
Various of plug-in	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：It can specify only the positive value for Surface Color, Luminosity, Diffuse Level, Specular Level, Glossiness. If it configure the negative value (such as -10%), it transfers as 0%.

※2：It corresponds ON/OFF configuration for Smoothing, the threshold is always transferred as 180℃.

Table5.2-d Texture function

Process function	Corresponding condition
Surface Color	◎
Luminosity	×
Diffuse Map(Diffuseness)	◎
Reflection Map(Specularity)	×
Specular Map(Reflectivity)	×
Transparency Map(Transparency)	×
Bump Map(Bumpiness)	×
Planar Image Map	◎
Cylindrical Image Map	◎
Spherical Image Map	◎
Cubic Image Map	◎
Other mapping	×
Counterturn of Layer	×
Texture Opacity	×
Blend Mode	×
Pixel Blending	×
Width Repeat	○ ※1
Height Repeat	○ ※1
Reference object	×
Texture Axis	◎
Texture Size	◎
Texture Center Position	◎
Texture Rotation Angle	×
Texture Falloff	×
Texture Anti-aliasing	○ ※2
World Coordinates	×
UVMap	◎

◎：Correspond, ○：Partially correspond, ×：Non-correspond

(Continued to next page)

※ 1：The configuration of the texture wrap is transferred in the following. Reset, Mirror is parameters, which is newly added in LightWave3D Ver.6.x. It happens to differ from the drawing result of LightWave3D for Reset.

Repeat → REPEAT

Edge → CLAMP

Reset → It is processed as BORDER(BORDER COLOR configures color) in 3DGL scene drawing API.

Mirror → Non-correspond(transferred as REPEAT)

※2：Texture anti-aliasing corresponds only Enable/Disable. The strength does not correspond.

Table5.2-e Image function

Process function	Corresponding condition
Load Image Sequence(Load Sequence)	×
Load Flyer Clip	×
Replace Image	×
Loop Image Sequence	×
Sequence Loop Length	×
Change Color of Image Pallet(Color Cycling)	×
Specify Starting Pallet Position(Low Cycle Index)	×
Specify Ending Pallet Position(High Cycle Index)	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

Table5.2-f Light function

Process function	Corresponding condition
Ambient Color	◎
Ambient Intensity	◎
Ambient Envelope	○ ※1
Light Color	◎
Light Intensity	◎
Light Envelope	○ ※1
Light Type	○ ※2
Light Intensity Falloff	×
Light Intensity Falloff Envelope	×
Spotlight Cone Angle	○ ※3
Spotlight Cone Angle Envelope	×
Spot Soft Edge Angle	×
Spot Soft Edge Angle Envelope	×
Enable/Disable of the diffuse level(No Diffuse)	×
Enable/Disable of the reflection(No Specular)	×
Light Projection Image	×
Lens Flare Related Function	×
Shadow Map Related Function	×

◎：Correspond, ○：Partially Correspond, ×：Non-correspond

※1：It does not correspond to the dynamic configuration of the brightness by the envelope. When it configures the brightness using the envelope, only the initial configured value is transferred.

※2：It does not correspond to the line light, and the face light, which are added from Ver.6.x.

※3：The configurable corn angle is between 0℃~90℃. If the angle, less than 0℃ or above 90℃, is configures, the value is clamped less than 0℃ to 0℃, above 90℃ to 90℃.

Table5.2-g Camera function

Process function	Corresponding condition
Rendering Type	×
Basic Resolution	◎
Multiple Number of Resolution	×
Optional Configuration of Resolution(Custom Size)	◎
Aspect Ratio of Pixel(Pixel Aspect Ratio)	○ ※1
Limited Region	×
Segment Memory	×
Anti-aliasing	×
Soft Filter	×
Adaptive Sampling	×
Sampling Threshold	×
Zoom Factor	◎
Zoom Factor Envelope	×
NTSC Wide Screen	×
Film size, Aperture Height(Film Size)	○ ※2
Field Rendering	×
Motion Blur	×
Blur Length	×
Particle Blur	×
Dithered Motion Blur	×
Depth of Field	×
Focal Distance	×
FOV Horizontal	×
FOV Vertical	×
Lens F-Stop	×
Stereo/DOF	×
Mask	×
Ray trace Related function	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：Aspect ration of the pixel is only “Square Pixels” in Ver5.x and “1.0” in Ver6.x corresponding.

※2：The Film Size, Aperture Height corresponds to only “35mm motion picture”.

Table5.2-h Effect(special effect)function

Process function	Corresponding condition
Back Ground Color(Backdrop Color)	×
Fog Type	×
Image Composting	○ ※1
Glow Effect	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：It corresponds to only the back ground composition.

Table5.2-i Record function

Process function	Corresponding condition
Automatic Progressive	×
Frame Process Ending Buzzer	×
Render process Progression Display	×
Render Display Destination	×
VIPER Enable	×
Rendering ·Tab and following	×
Output File ·Tab and following	×
Device Control ·Tab and following	×

◎ ：Correspond, ○：Partially correspond, ×：Non-correspond

Table5.2-j Object Motion Function

Process Function	Corresponding Condition
Movement	○ ※1
Rotation	○ ※1
Scaling Up/Down	○ ※1
Stretch	×
Movement of Center Point	×
Rotation of Center Point	×
	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：It corresponds to only the straight line and the constant.

Table5.2-k Light motion function

Process function	Corresponding condition
Movement	○ ※1
Rotation	○ ※1
Movement of Center Point	×
Rotation of Center Point	×
Bus Movement	×
Corn Angle	○ ※2

◎：Correspond, ○：Partially correspond×：Non-correspond

※1：It corresponds to only the straight line and the constant.

※2：It corresponds only in using the spot light. It does not correspond to the dynamic configuration of the corn angle by the envelope.

Table5.2-1 Camera motion function

Process function	Corresponding condition
Movement	○ ※1
Rotation	○ ※1
Movement of Center Point	×
Rotation of Center Point	×
Bus Movement	×
Camera Zoom	○ ※2

◎：Correspond, ○：Partially correspond, ×：Non-correspond

※1：It corresponds to only the straight line and the constant.

※2：It does not correspond to the dynamic configuration of the camera zoom by the envelope.

Table5.2-m Motion option function

Process function	Corresponding condition
Parent Object	×
Target Object	×
IK	×
Control and Restriction	×

◎：Correspond, ○：Partially correspond, ×：Non-correspond

6. START-UP

6.1. Execution mode

The model converter is performed by either GUI mode or the command mode.

The execution format file is differing in each mode. It shows the execution format file name of each execution mode in Table6.1.

Table6.1 Execution mode and execution format file name of the model converter

Execution mode	Execution format file name
GUI mode	lwmdfw.exe
Command mode	lwmdf.exe

6.2. GUI mode

GUI mode performs the transfer process using the operation panel.

6.2.1. How to start GUI mode

In GUI mode, it starts by directly executing lwndfw.exe or creating the short cut.

When it executes by creating the short cut, it can specify the option and input file from the property of the short cut.

Format

```
lwmdfw  [/Option[ ...]] [Input file[ ...]]
```

Option

If it specifies the option, the value is configured in the operation panel in starting up the model converter. For the detail of the option, refer to *6.4 Start up option*.

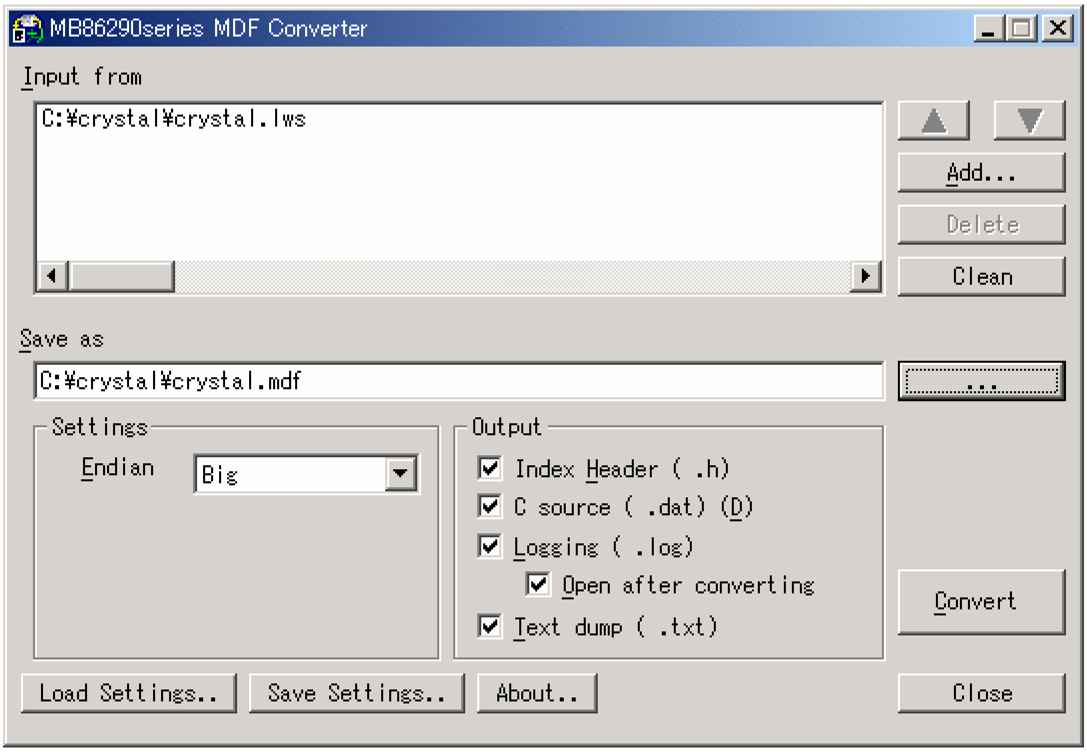
Input file

It can specify the scene file, the object file, the image data file, and the extension model data file for the input. Also, it can specify the input file on the operation panel after starting up. Refer to *Table4.1-a Input file* for the detail.

6.2.2. Operation panel of GUI mode

If it executes lwmdfw.exe, the following operation panel (GUI mode operation display) show up.

In the following, it explains each item of GUI mode operation display.



Input from
C:¥crystal¥crystal.lws

Save as
C:¥crystal¥crystal.mdf

Settings
Endian Big

Output
☑ Index Header (.h)
☑ C source (.dat) (D)
☑ Logging (.log)
☑ Open after converting
☑ Text dump (.txt)

Input from

Input file list configured currently is displayed.

▼Button

It steps down the file order, which is selected by the input file list.

▲Button

It steps up the file order, which is selected by the input file list.

Add... button

It adds the input file to the input file list.

The following file could be specified for the input file.

- Scene file (.lws)
- Object file (.lwo)
- Texture image
    - o Windows bit-map (.bmp)
    - o IFF format (.iff)
    - o TGA format (.tga)
- Extension model data file (.mdx)

Delete button

It deletes the file from the list, which is selected by the input file list.

Clean button

It deletes all files, which is configured in the input file list.

Save as

It specifies the model data file for the output. It can be specified in the following file format.

- Model data format ([model data file name].mdf)
- Extension model data format ([model data file name].mdx)

Endian

It specifies the endian.

If it specify the model data format as the output file, it is necessary to specify the endian.

If it specify the extension model data format as the output file, it is not necessary to specify the endian.

Index Header (.h) check button

When the check button is ON, it output the model index file, named [model data file name].h. The model data index file is the header file, which defines the index number.

C source (.dat) check button

When the check button is ON, it output the file, named [model data file name].dat. This file is the model data is described in source program format of C language.

Logging (.log) check button

When the check button is ON, it output the convert log file, named [model data file name].log. The convert log file is a file, which the condition of the transfer process is written.

Open after converting check button

When the check button is ON, it shows the output convert log file by NotePad.

Text dump (.txt) check button

When the check button is ON, it output [model data file name].txt file. This file is a text file, which the content of the model data is transferred.

Load Settings.. button

It loads the configuration in the transfer from the configuration file (.mcs). It can create the configuration file by Save Settings.. button.

Save Settings.. button

It saves the input file, which is displayed on GUI mode operation display, the output file, the configuration of the endian on the configuration file (.mcs). The saved content can be recovered by the Load Settings.. button. It is convenient to create the configuration file in case of repeating an identical transfer operation.

About.. button

It is a copyright notice of the model converter.

Convert button

It starts the transfer process.

If the necessary parameter for the transfer process is not configured, it can not press the button.

Close button

It terminates the model converter.

If it is necessary to save the configured content, it pushes Save Settings.. button before pushing Close button, and then saves on the configuration file.

6.3. Command mode

The command mode performs the transfer process by the command prompt.

Format

```
lwmdf  [/Option[ ...]] /O:Output file  [Input file[ ...]]
```

Option

Refer to *6.4 Start up option* for the detail of the option.

Output file

It specifies the creating model file or the extension model data file.

It is necessary to specify the output file by the command line. If it specifies *the configuration file* by the option, it can not use the output file information inside the configuration file.

Input file

It can specify the scene file to input, the object file, the image data file, and the extension model data file.

6.4. Start up option

It can specify the start up option in starting up. It shows the start up option in Table6.4.

Table6.4 Classification and content of the start up option

Option	Content	Default
/O:output-file	It specifies the output file name. It can specify either the model data format (.mdf) or the extension model data format (.mdx), which can be re-input.	If it does not specify the extension on the output file name, it is output by the model data format (.mdf).
/S:setting-file	It specifies the configuration file to load in starting up the model converter. The configuration file is a file, which save the configuration of the input file in transferring, the output file, and the endian. The model converter automatically creates the configuration file, named [file name]current.mcs in transfer process. Additionally, it can create the configuration file in any file name by Save Settings.. button of *6.2.2 GUI mode operation display*. If it specifies the input file in the command line to start up the model converter, it ignores the input file, described in the configuration file.	If it does not specify this option, it treats current.mcs as the configuration file. Also, if it does not specify the extension in the configuration file name, (.mcs) becomes the extension.
/TOOL:tool-path	It specifies the install path of LightWave3D.	If it does not specify this option, C:¥LightWave becomes the install path.
/BIG	It changes the endian to BIG or LITTLE.	GUI mode： If /BIG or /LITTLE is not specified, the endian configuration in starting up becomes the un-configuration condition.
/LITTLE		Command mode： If it does not specify this option, it follows the content described in the configuration file.

(to be continued)

Table6.4 Classification and content of the start up option (continued)

Option	Content	Default
/HEADER	It controls the output of the index header file.	If it does not specify this option, it follows the content described in the configuration file.
/NOHEADER		
/DATA	It controls the output of he source program file described in C language of the model data.	
/NODATA		
/DUMP	It stops to output the text dump file.	
/NODUMP		
/LOG	It controls the output of the log file. If it specifies NOLOG in the command mode, it output the content of log on the console.	
/NOLOG		
/HELP	It displays the help message by the command mode. It does not perform the transfer process. (only command mode)	—

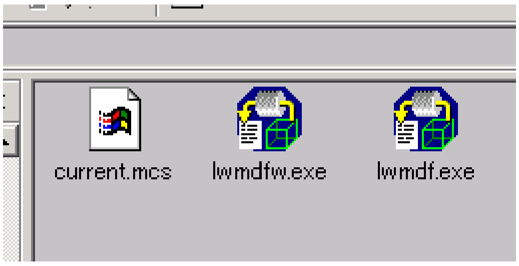
【**Supplement**】

- It can not specify the output file and the configuration file more than once.
- If the conflicting option (/BIG and /LITTLE, or /HEADER and /NOHEADER) is specified, the option, which is specified last, is enable.
- It can use each option regardless of the big letter and the little letter.

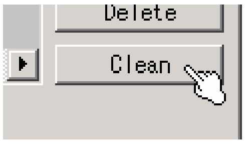
7. OPERATION PROCEDURE

7.1. It creates the model data by GUI mode.

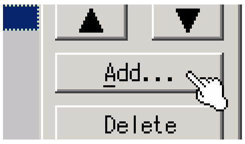
(1) Execute lwmdfw.exe.



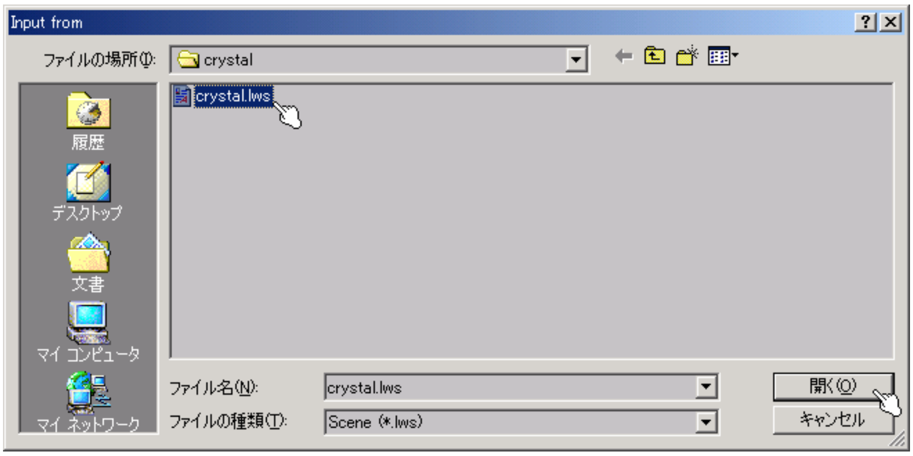
(2) If there is a file in the list box, Input from, it press Clean button.



(3) Push Add button.



(4) After it confirmed the type of the file as (*.lws), it specifies the scene file for the transfer, and pushes *open* button.


Input from

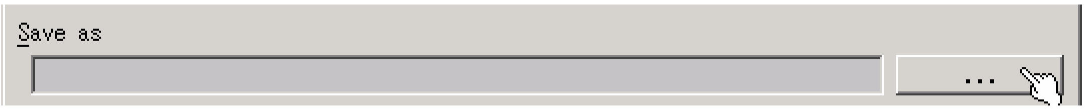
ファイルの場所(I): crystal

crystal.lws

ファイル名(N): crystal.lws

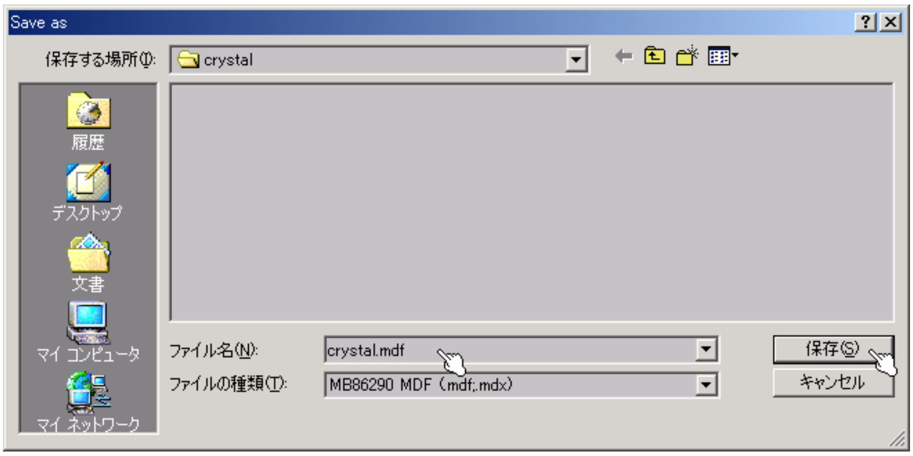
ファイルの種類(T): Scene (*.lws)

(5) Pushes [..] next to the edit control of [Save as].



Save as

(6) Move the folder to output, input the file name of the model data, and push [Save] button.



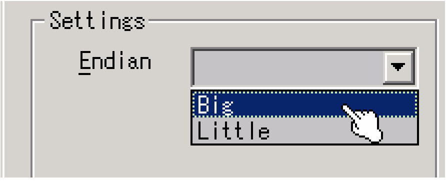
Save as

保存する場所(I): crystal

ファイル名(N): crystal.mdf

ファイルの種類(T): MB86290 MDF (mdf;.mdx)

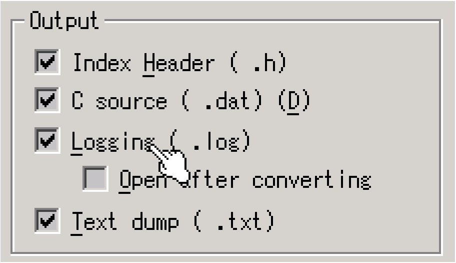
(7) Select Endian from [Endian] list of [Settings]. Select the endian corresponding the target CPU (or its mode).



Settings

Endian

(8) Select the output file, except the model data by [Output].



Output

☑ Index Header (.h)

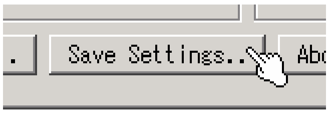
☑ C source (.dat) (D)

☑ Logging (.log)

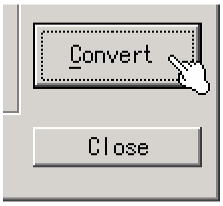
□ Open after converting

☑ Text dump (.txt)

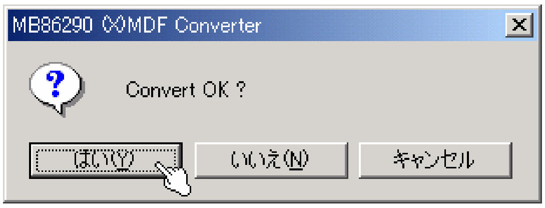
(9) If it needs to repeat this configuration later, push [Save Settings] button, and save the configuration file. The extension of the configuration file is .mcs.



(10) If all configuration is done, push [Convert] button.

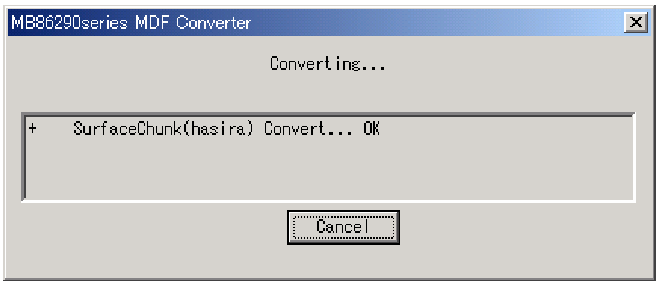


(11) The configuration message, [Convert OK?] is displayed, and push [OK] button.


MB86290 (X)MDF Converter

Convert OK ?

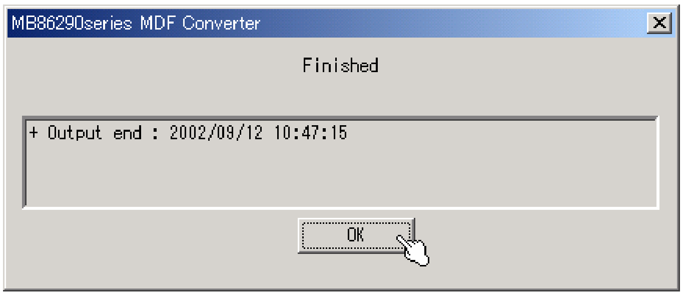
(12) It starts the transfer process. If it terminates the transfer, push [Cancel] button in transferring.


MB86290series MDF Converter

Converting...

```
+    SurfaceChunk(hasira) Convert... OK
```

(13) After finishing the transfer, [Finished] is displayed, and push [OK] button.

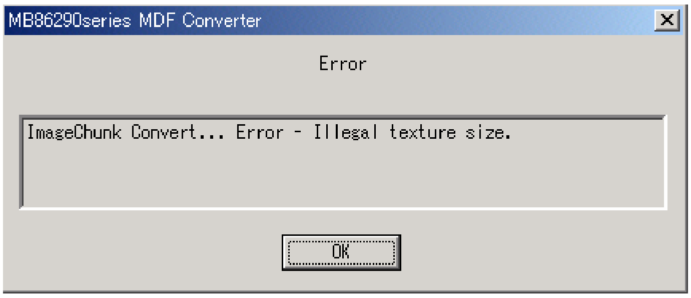


```
+ Output end : 2002/09/12 10:47:15
```

(14) If the error during the transfer occurs, it outputs the following error message and terminates the transfer.

Confirm the content of the error, and modify the scene.

Refer to *9 Error message* for the content of the error message.

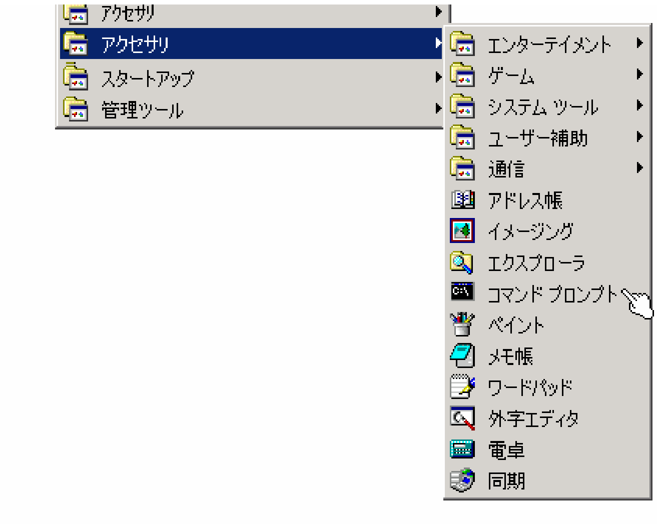


```
ImageChunk Convert... Error - Illegal texture size.
```

7.2. Create the model data on the command mode

In advance, it copies lwmdf.exe to the folder, which the environmental variable PATH is configured, or adds the environmental variable PATH to the folder, which lwmdf.exe is stored.

(1) Start up the command prompt.



(2) Input the command from the command prompt.

First, move to the folder, which the scene file is stored.

The following example is the case, the scene file is stored in the crystal folder under C drive.

```
C:¥>cd ¥crystal
C:¥crystal>
```

(3) Next, input the command, and transfer it. In the following example, it transfers crystal.lws by BIG endian and creates crystal.mdf. After finishing the transfer, the prompt is displayed again following the display, MB86290series 3D Model•••.

```
C:¥crystal>lwmdf /o:crystal.mdf /big crystal.lws
MB86290series 3D Model Data Converter for LightWave V02L01R01
ALL RIGHTS RESERVED, COPYRIGHT (C) FUJITSU LIMITED 2002
LICENSED MATERIAL - PROGRAM PROPERTY OF FUJITSU LIMITED

C:¥crystal>
```

8. OUTPUT FILE

8.1. Model data file（.mdf）

It shows the rough organization of the model data file in Figure8.1. As shown in Figure8.1, the model data is organized in the data division called the chunk. In the following, it explains each division.

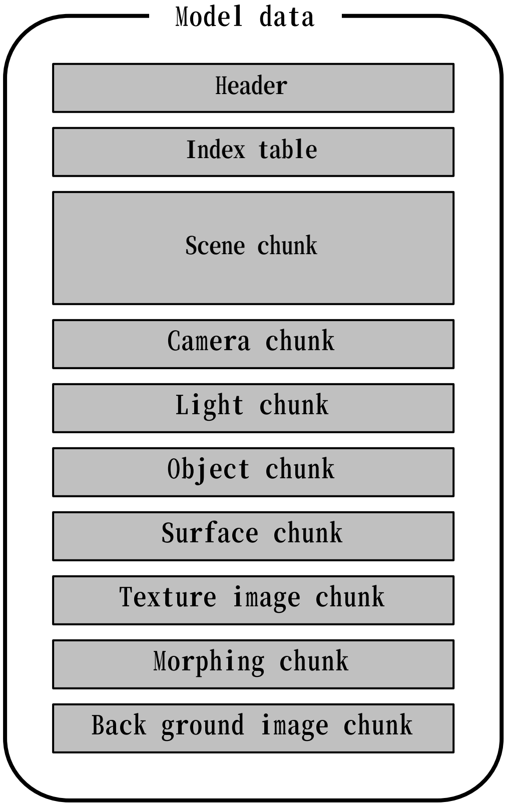


Figure8.1 Organization of model data

■**Header**

The header is the data area, which stores the object in the model data, the number of the surface, and the version information of the model data file.

■**Index table**

The index table is the pointer array area, which access each chunk.

■**Scene chunk**

The scene chunk is the area, which stores a camera, a light, an object, the pointer to the back ground information used in the scene, and its motion information.

■ **Camera chunk**

It stores the camera information.

■ **Light chunk**

It stores the light information.

■ **Object chunk**

It stores the object information. The object chunk includes the pointer information to the object chunk, the surface chunk, and the texture image chunk.

■ **Surface chunk**

It stores the data, organizes the surface. The surface data is composed of the apex coordinate, the normal vector, and the texture coordinate.

■ **Texture image chunk**

It stores the image data to use the texture mapping.

■ **Morphing chunk**

It stores the morphing information included in the scene.

■ **Back ground image chunk**

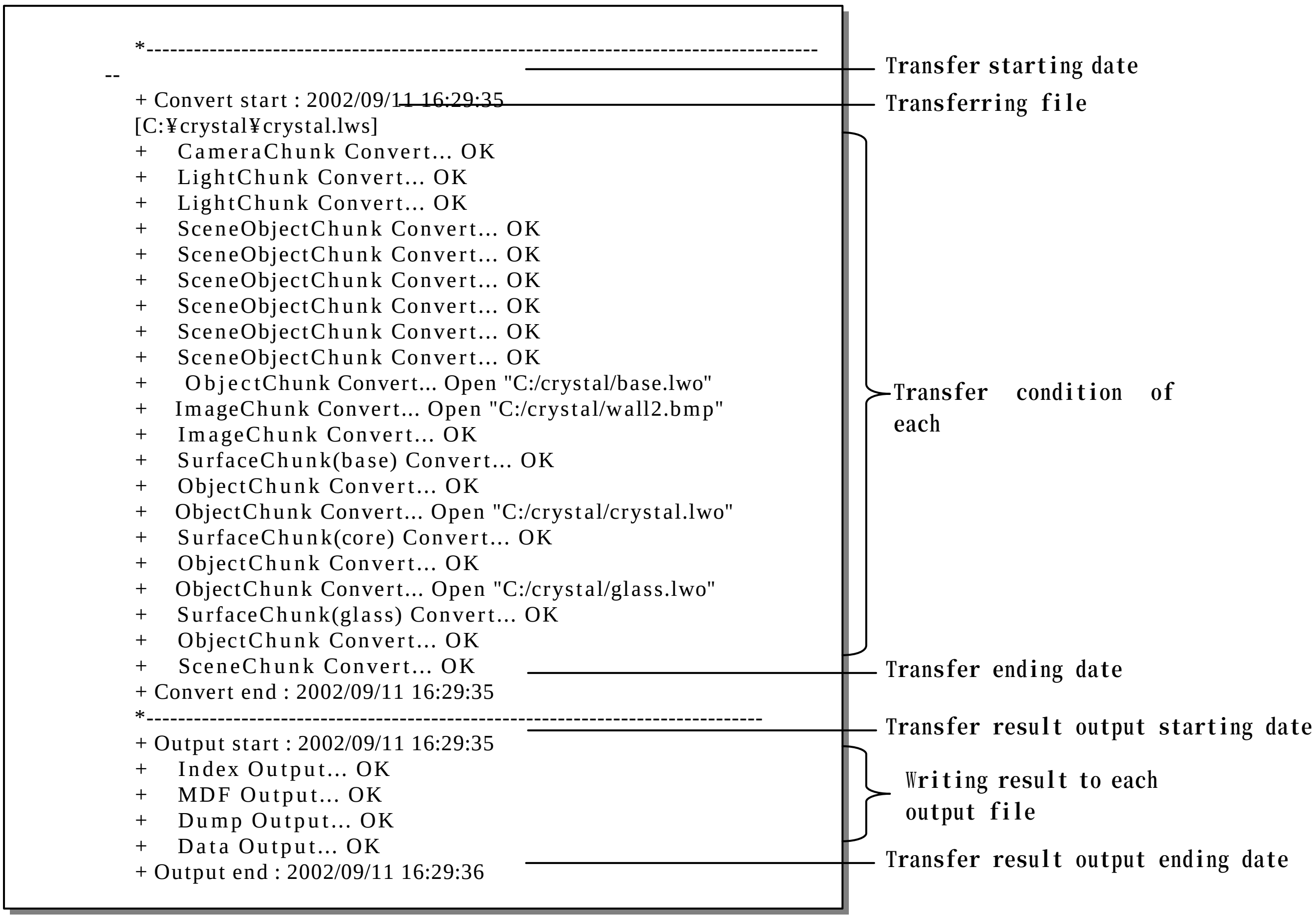
It stores the back ground image of the scene. In the model data, it can treat one back ground image data for one scene.

8.2. Extension model data file（.mdx）

It is the extension file, which can re-organize the model data. The extension model data file is not treated as the scene data in 3DGL scene drawing API, it can be used the input data again in the model converter.

8.3. Convert log file（.log）

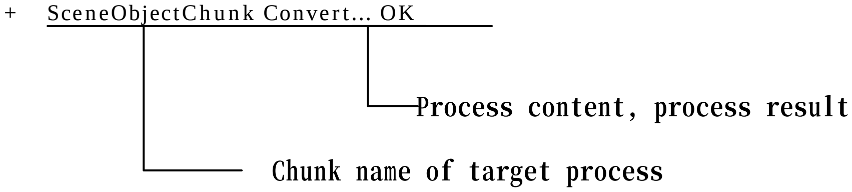
After finishing the transfer process, the process condition is written on the convert log file, named [model data file name].log. In the following, it explains an example of the convert log file and its content. If an error occurs or a warning occurs, it output as the log information. Refer to *9 List of the error message* and *10 List of the warning message* for the content of an error message and a warning.



```
*-----------------------------------------------------------------------------------
--
+ Convert start : 2002/09/11 16:29:35
[C:¥crystal¥crystal.lws]
+   CameraChunk Convert... OK
+   LightChunk Convert... OK
+   LightChunk Convert... OK
+   SceneObjectChunk Convert... OK
+   SceneObjectChunk Convert... OK
+   SceneObjectChunk Convert... OK
+   SceneObjectChunk Convert... OK
+   SceneObjectChunk Convert... OK
+   SceneObjectChunk Convert... OK
+    ObjectChunk Convert... Open "C:/crystal/base.lwo"
+   ImageChunk Convert... Open "C:/crystal/wall2.bmp"
+   ImageChunk Convert... OK
+   SurfaceChunk(base) Convert... OK
+   ObjectChunk Convert... OK
+   ObjectChunk Convert... Open "C:/crystal/crystal.lwo"
+   SurfaceChunk(core) Convert... OK
+   ObjectChunk Convert... OK
+   ObjectChunk Convert... Open "C:/crystal/glass.lwo"
+   SurfaceChunk(glass) Convert... OK
+   ObjectChunk Convert... OK
+   SceneChunk Convert... OK
+ Convert end : 2002/09/11 16:29:35
*---------------------------------------------------------------------------
+ Output start : 2002/09/11 16:29:35
+   Index Output... OK
+   MDF Output... OK
+   Dump Output... OK
+   Data Output... OK
+ Output end : 2002/09/11 16:29:36
```

Transfer starting date
Transferring file
Transfer condition of each
Transfer ending date
Transfer result output starting date
Writing result to each output file
Transfer result output ending date

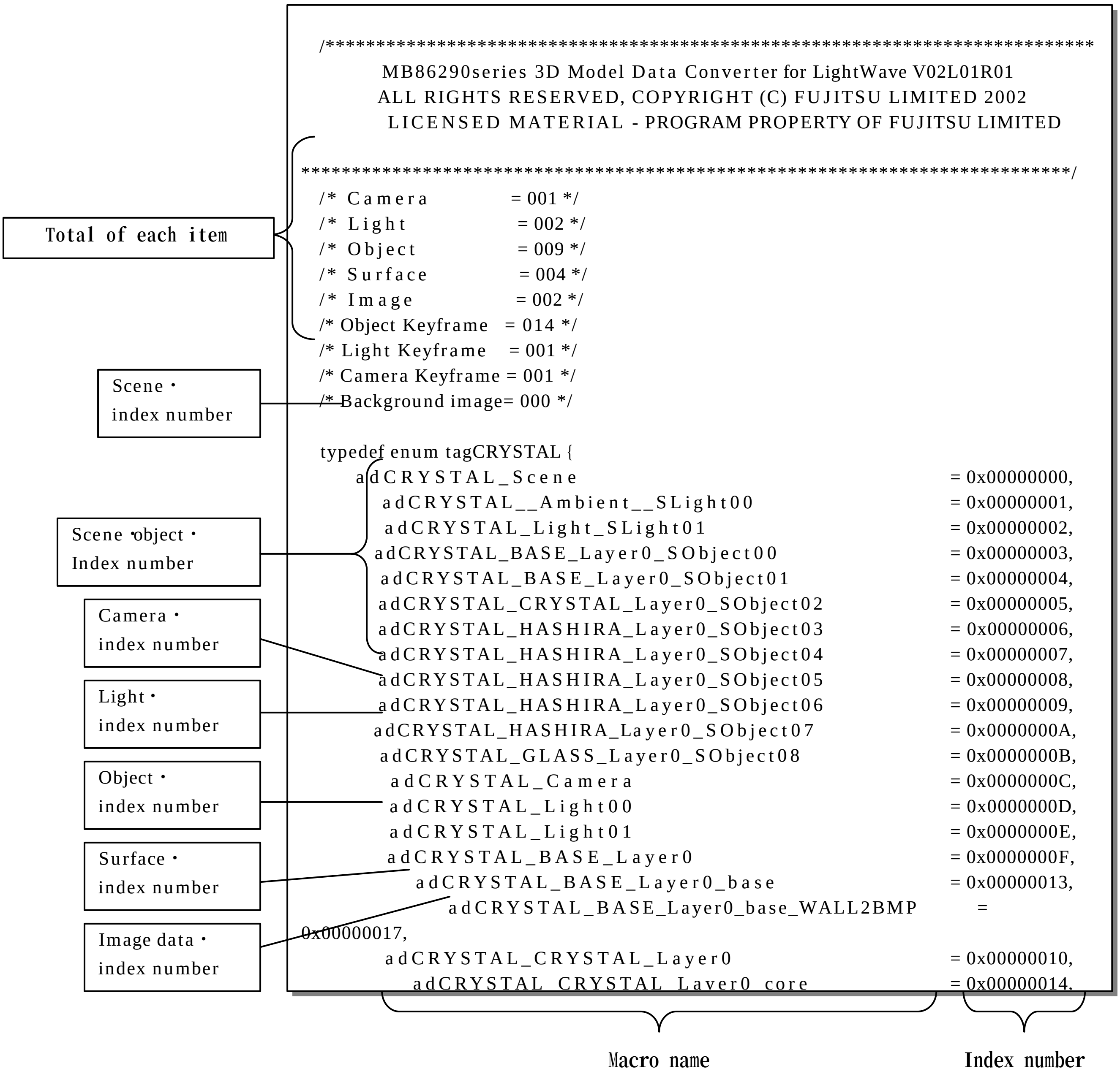
※1 【Output format】



```
+   SceneObjectChunk Convert... OK
```

8.4. Model data index file（.h）

After finishing the transfer process, it creates the model data index file named [model data file name].h in the same folder as the model data file. In the model data index file, the index number is described as the macro to access the model data file from the user application. The macro is regularly named based on each item name in LightWave3D. In the user application, it specifies any item in this macro, and uses 3DGL scene drawing API. It shows an example of the model data index file in the following. Also, it shows the index classification in Table8.4.1 and the naming rules of the macro in Table8.4.2.



```
/*****************************************************************************
        MB86290series 3D Model Data Converter for LightWave V02L01R01
        ALL RIGHTS RESERVED, COPYRIGHT (C) FUJITSU LIMITED 2002
         LICENSED MATERIAL - PROGRAM PROPERTY OF FUJITSU LIMITED

*****************************************************************************/
/* Camera            = 001 */
/* Light             = 002 */
/* Object            = 009 */
/* Surface           = 004 */
/* Image             = 002 */
/* Object Keyframe   = 014 */
/* Light Keyframe    = 001 */
/* Camera Keyframe = 001 */
/* Background image= 000 */

typedef enum tagCRYSTAL {
    adCRYSTAL_Scene                                           = 0x00000000,
      adCRYSTAL__Ambient__SLight00                            = 0x00000001,
      adCRYSTAL_Light_SLight01                                = 0x00000002,
     adCRYSTAL_BASE_Layer0_SObject00                          = 0x00000003,
      adCRYSTAL_BASE_Layer0_SObject01                         = 0x00000004,
      adCRYSTAL_CRYSTAL_Layer0_SObject02                      = 0x00000005,
      adCRYSTAL_HASHIRA_Layer0_SObject03                      = 0x00000006,
      adCRYSTAL_HASHIRA_Layer0_SObject04                      = 0x00000007,
      adCRYSTAL_HASHIRA_Layer0_SObject05                      = 0x00000008,
      adCRYSTAL_HASHIRA_Layer0_SObject06                      = 0x00000009,
     adCRYSTAL_HASHIRA_Layer0_SObject07                       = 0x0000000A,
      adCRYSTAL_GLASS_Layer0_SObject08                        = 0x0000000B,
       adCRYSTAL_Camera                                       = 0x0000000C,
       adCRYSTAL_Light00                                      = 0x0000000D,
       adCRYSTAL_Light01                                      = 0x0000000E,
       adCRYSTAL_BASE_Layer0                                  = 0x0000000F,
          adCRYSTAL_BASE_Layer0_base                          = 0x00000013,
             adCRYSTAL_BASE_Layer0_base_WALL2BMP                 =
0x00000017,
       adCRYSTAL_CRYSTAL_Layer0                               = 0x00000010,
          adCRYSTAL_CRYSTAL_Layer0_core                       = 0x00000014,
```

Table8.4.1 Classification of index

Classification	Description
Scene ·index number	It distinguishes each scene. It is assigned one for each scene.
Object ·index number	It distinguishes each object in the model data. It is assigned one for each object in the model data.
Scene ·object ·index number	It distinguishes an object in the scene. The difference from the object ·index number is it is associated with the scene. For example, if there are multi object in the scene or same object in the multi scene, the object ·index number is same, but the different object ·index number is assigned. There is no scene ·object ·index number for the object, which is not including in the scene.
Surface ·index number	It distinguishes each surface, which makes up the object. It assigns one for each surface.
Image data ·index number	It assigns the image for the texture mapping and the image for the back ground. It assigns one for each image.
Camera ·index number	It distinguishes a camera. However, there is only one camera for each scene in 3DGL scene drawing API, and the camera ·index number is not used.
Light ·index number	It distinguishes a light. However, in 3DGL scene drawing API, it uses the light number as same as the core API for specifying the light, the light ·index number is not used.

Table8.4.2 Naming rule of the macro

■**Scene･index number**

ad scene file name_Scene

■**Object･index number**

ad scene file name_object file name_Layer#

■**Scene･object･index number**

ad scene file name_object file name_Layer#_SObject##

■**Surface･index number**

ad scene file name_object file name_Layer# _surface name

■**Image data･index number**

ad scene file name_object file name_Layer#_surface name_image file name

■**Camera･index number**

ad scene file name_Camera

■**Light･index number**

ad scene file name_Light##

■**Object･index number**(in case of an object itself being transferred)

ad_object file name_Layer#

■**Image data･index number**(in case of a image data itself being transferred)

ad__image file name

- ✧ The image file name is described as [ABCBMP] for [ABC.BMP].
- ✧ Layer# is a layer number, which starts from 0.
- ✧ SObject## is a sequential number of the scene･object information, which starts from 00. (If a object itself is transferred, there is no index.)
- ✧ Light## is a sequential number of the light information, which starts from 00.

9. ERROR MESSAGE LIST

If an error occurs, it terminates the transfer process. There are following error messages.

【A】

Abort by user.

The process is aborted by the user.

【C】

Cannot open data. (file name)

It can not open a data file.

Cannot open for logging. (file name)

It can not open a log file.

Cannot open image. (image name)

It can not open an image.

Cannot open object. (object name)

It can not open an object.

Cannnot open MDF. (file name)

It can not open MDF(model data file).

Cannot open scene. (scene name)

It can not open the scene.

Cannot open text. (file name)

It can not open a text dump file.

Cannot read from MDF.

It can not read MDF(model data file).

Cannot write to data.

It can not write to a data file.

【C】

Cannnot write to logging file.

It can not write to a log file.

Cannot write to MDF.

It can not write to MDF(model data file).

Cannot write to text.

It can not write to a text dump file.

Cannot write to index file.

It can not write to a index file.

Child object not support.

It does not support a child object.

【D】

Duplicate clipmaps.

Multi clip maps are specified on one texture image.

【I】

Illegal BMP.

This format can not treat an image of BMP format.

Illegal clipmap size.

The size of a texture image and a clip map do not match.

Illegal IFF.

This format can not treat an image of IFF format.

Illegal image compress.

It can not treat a compress image.

【I】

Illegal image size.

The width or height of the back ground image is not the multiple number of four.

Illegal image type.

This is an illegal image format.

Illegal target object.

There is an error specifying the morph target object.

Illegal texture size.

It can not treat the size of the texture.

Illegal TGA.

It can not treat TGA format image.

Illegal wrap mode.

There is an error on the texture wrap mode.

Illegal XMDF format.

It can not treat as XMDF(extension model data format).

Internal error.

It is an internal error. Ask for a distribution source.

Invalid camera type.

A camera for a target is configured.

Invalid frame No. or step.

The negative value on the starting frame, ending frame, or step frame is configured.

【M】

Material is not assigned.

The material on the object is not assigned.

【N】

Not scene, object or image. (file name)

Those files can not perform the transfer process.

【O】

Out of memory.

It can not continue the process due to out of memory.

【P】

Please specify .mdf or .mdx as saving file.

Specify a file, which has the extension of .mdf or .mdx for the output file.

【T】

Texture coordinate is not assigned.

The texture coordinate is not assigned.

Texture image is not set (surface name).

The texture image is not specified on the surface, which has a texture property.

Too many BG images.

There are too many back ground image.

Too many cameras.

There are too many cameras.

Too many images.

There are too many texture images.

Too many lights.

There are too many lights.

Too many morphings.

There are too many morphing target object.

【T】

Too many motion keys.

There are too many motion key.

Too many objects.

There are too many objects.

Too many scenes.

There are too many scenes.

Too many scene BG images.

There are too many scene back ground images.

Too many scene cameras.

There are too many cameras.

Too many scene lights.

There are too many scene lights.

Too many scene objects.

There are too many scene objects.

Too many surfaces.

There are too many surfaces.

Too many texture vertices.

There are too many texture apexes.

10. WARNING MESSAGE LIST

Even if the warning occurs, the transfer process continues. There are following warning message.

【B】

Background image can use only one in a scene."

More than two back ground images are configured. (only one back ground could be transferred)

【C】

Camera aspect ratio is not 1:1.

The camera aspect ration is not 1:1.

Camera can use only one in a scene.

More two cameras are configured. (only one camera could be transferred)

【I】

Image (texture name) width or height is over 256 pixel.

The width or height of the texture image is over 256 pixels.

Image (texture name) width or height is over 4096 pixel.

The width or height of the texture image is over 4096 pixels. (only Coral)

Invalid light type.

There are some invalid light types. (It does not transfer invalid light type.)

【T】

Texture coordinate is out of height range -512 to +511.

The texture V coordinate is out of -512~+511.

Texture coordinate is out of height range -8192 to +8191.

The texture V coordinate is out of -8192~+8191. (only Coral)

Texture coordinate is out of width range -512 to +511.

The texture U coordinate is out of -512~+511.

【T】

Texture coordinate is out of width range -512 to +511.

The texture U coordinate is out of -512~+511.

Texture coordinate is out of width range -8192 to +8191.

The texture U coordinate is -8192~+8191. (only Coral)

Texture wrap mode 'mirror' was changed to 'repeat'.

“mirror” is configured in the texture wrap mode. It change to “repeat”.

The number of the light over 8.

The configured lights are over eight. (can transfer up to eight)

The spot cone angle is set to 0.

As the corn angel of the spot light is under 0℃, it re-configure 0℃.

The spot cone angle is set to 180.

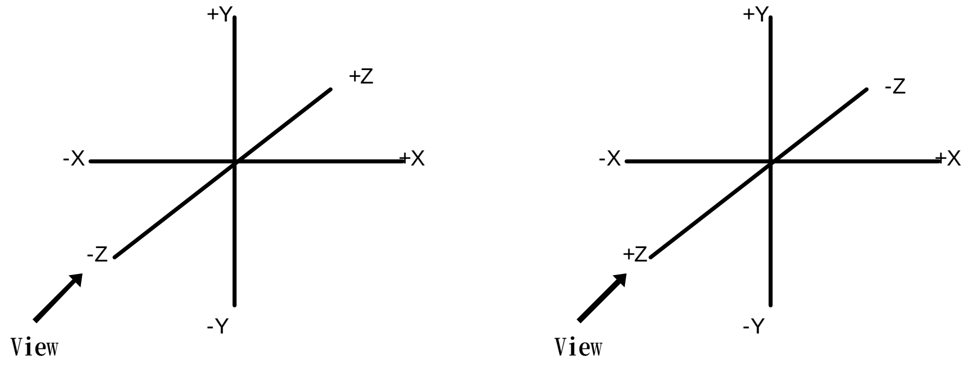
As the corn angle of the spot light is over 90℃, it re-configure 180℃.

(color value) value was clipped to minimum or maximum.

It clips the color value to 0~255.

11. RESTRICTIONS

- It does not guarantee the transfer result, when it uses except alpha-numeral for the surface name.
- The image data size, which can be transferred, is depended on the graphics controller. (refer to Appendix A)
- The back ground image, both of the width and height is multiple number of four, and the maximum is width 4096 x height 4096.
- It can not transfer the back ground image with the scaling up/down. Specify the same size as the back ground image size.
- It can transfer only one camera.
- The projection value (Near, Far) on the camera projection process is configured as (1, 200) regardless of the configuration of the modeling tool. Refer to 3DGL scene drawing API in changing the value.
- The threshold of the smoothing is processed as 180℃ regardless of the configuration of the modeling tool.
- The image of the clip map must be two value data of the original image. On this occasion, the configuration of the original image is used for the image size, center position, mapping method, wrapping of the parameter. As the parameter configured in the clip map image is valid in LightWave 3D, the clip map process of the model converter compares the original image with the clip map image, and creates the original image, which configure the stencil bit (The converter transfers to 1-5-5-5 format. This is, one pixel is composed of 1+5+5+5=16 bits, MSB is a stencil bit, another bits are red, green, and blue of each 5 bits.). Therefore, the parameter configured in the clip map is invalid.
- If it performs the morphing with the texture, the morphing destination object must be as same as the image configuration of the original object, because when it performs the interpolating calculation of the texture coordinate, it needs the texture coordinate of the morphing destination texture coordinate. If it does not perform the image configuration, it does not output the texture coordinate, and does not perform normally the morphing.
- The terminal light color is clamped to each RGBA 0~255. Therefore, the light, which has a negative strength (such as Intensity is -20%), is transferred to 0.
- The terminal material color of the surface is clamped to each RGBA 0~255. Therefore, the material color, which has a negative strength, is transferred to 0.
- As shown in Figure11.1, there is difference between the configuration method of the world coordinate in LightWave3D and the world coordinate in 3D graphics library. Therefore, the value is transferred for 3D graphics library and output.



(A)World coordinate of LightWave 3D

(B)World coordinate of 3D graphics library

Figure11.1 World coordinate system

As shown above, there is difference in Z-axis direction, and the view direction also differs. Therefore, in order to solve this differences, the configuration value of a camera and the projection value is changed in the following in the model converter.

- Camera rotation value (It adds 180 to H-axis and reverse the sign, and adds 180 to B-axis.)

  Rot_H = −(H value of LightWave + 180.0)

  Rot_P = P value of LightWave 3D

  Rot_B = (B value of LightWave3D + 180.0)

- Projection value (Bottom value and Top value is calculated in the following.)：

  Bottom = (1.0/ZoomFactor) × (Width/Height)

  Top = − (1.0/ZoomFactor) × (Width/Height)

  Width: Camera resolution width LightWave

  Height: Camera resolution height LightWave

  ZoomFactor: Camera zoom factor of LightWave 3D

Also, LightWave 3D is configured in the world coordinate system, and 3D graphics library is configured in the camera coordinate system. The configuration of the world coordinate system is the position coordinate of a camera itself as same as a light, an object, however the configuration of the camera coordinate system is the position coordinate, which looks from a camera. Therefore, the movement value of a camera performs a sign change.

Tran_X = −(X coordinate of LightWave 3D)

Tran_Y = −(Y coordinate of LightWave 3D)

Tran_Z = −(Z coordinate of LightWave 3D)

A light and an object could be moved and rotated in the world coordinate of LightWave 3D due to these change. However, if it adds and changes the motion information of a camera in user application, it is necessary to consider this difference in configuring.

- The maximum lights to transfer are eight. The light number, 1~8 is assigned in order of configuring. If more than nine light is configured, eight lights is transferred in order of configuring and others lights are ignored. The ambient light of the full view is transferred as the light number 0.
- The corn angle of the spot light is configured between 0~90℃. If under 0℃ or above 90℃ is configured, it output the warning message and clamp to 0℃ and 180℃.
- The position and angel of the model data for MB86290 series in LightWave 3D might be differ depended on the specifying a light type. This difference is caused to resolve the different configuration between 3D graphics library and LightWave 3D, the value is transformed by the model converter, and this difference is not an error. Refer to Appendix E for the detail.
- It can transfer only the color map specified and Diffuse map specified out of the texture map. If other map is specified, it is ignored. However, do not specify except the color map specified and Diffuse map specified to keep away an error.
- The texture map, which can specify one surface, is either the color map or Diffuse map. If both of map are specified, the color map is prevailed, however specify either one to keep away an error.
- The texture image color, which is specified in Diffuse map of the modeling tool, is normally reflected on only the brightness, the color is reflected on the surface color, and however it output the texture image color in the converter. Therefore, in order to get the same result as specifying Diffuse map in the modeling tool, specify the texture image of the gray scale.
- There is a limit of the coordinate value for the wrap configuration of the texture map as shown below. If the texture coordinate, out of this limit, is configured, it does not normally perform the mapping.

  In the specification of the graphics controller, if the final texture coordinate value is between -512~+511 (-8192~+8191 in MB86293), the image is performed the wrapping.

  In the model converter, if the multiplication result of the texture coordinate (u, v) and the image size (w, h) is out of -512~+512, it output the warning message.
- The interpolating of the key frame in specifying the motion is performed by the linear interpolating. Therefore, even if the motion is specified with the curved line, it performs lineally.

Appendix-A The comparative chart of the graphics controller

Table A The comparison table of the graphics controller

		Graphics controller		
		MB86291(*1)	MB86292(*2)	MB86293 after (*3)
Maximum drawing frame size(*4)		4096×4096	4096×4096	4096×4096
Capacity of the graphics memory		2MB(internal)	Max 32MB	Max 32MB
Enable texture image size (*5) (width, height)	Using internal texture memory	4/8/16/32/64	4/8/16/32/64	
	Using graphics memory	4/8/16/32/64 /128/256	4/8/16/32/64 /128/256	4/8/16/32/64 /128/256/512/ /1024/2048/4096
Texture coordinate range		−512～511	−512～511	−8192～8191

*1)MB86921A、MB86291S included

*2)MB86292S included

*3)MB86293 and MB86294。

*4)The settable drawing frame size is within the capacity of the graphics memory.

*5)It can use the different size of the width and the height.

Appendix-B　Image file corresponding table

TableB Model converter corresponding image format

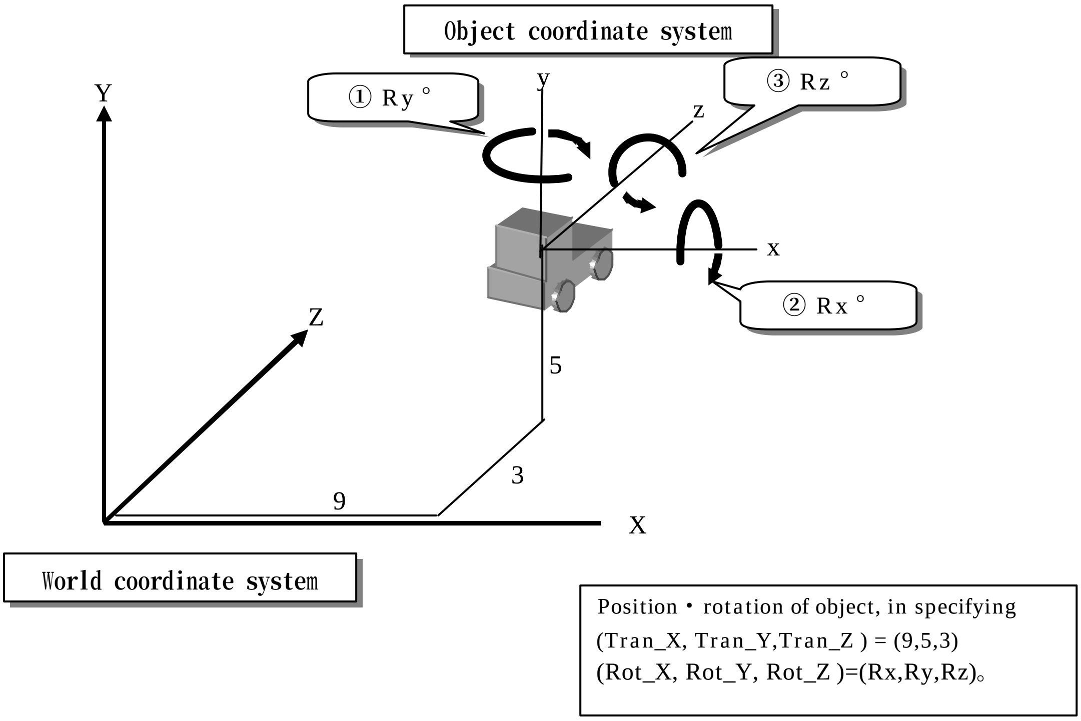
File format	*Compress/ Uncompress*	*Image bit length*
Windows bit map	Compress	4 bit
	Compress	8 bit
	Uncompress	8 bit
	Uncompress	24 bit
IFF	Compress	8 bit
	Compress	24 bit
	Uncompress	8 bit
TGA	Compress	8 bit
	Compress	16 bit
	Compress	24 bit
	Uncompress	8 bit
	Uncompress	16 bit
	Uncompress	24 bit

Appendix-C Change of the object motion

It explains the position and rotation of the object in adding and altering the object motion using 3DGL scene drawing API.

As shown in FigureC, if it specifies (Tran_X, Tran_Y, Tran_Z)=(9,5,3) as the position coordinate of the object, it put in position the origin of the object coordinate system for the coordinate (9,5,3) of the world coordinate system.

Also, if it specifies (Rot_X, Rot_Y, Rot_Z)=(Rx,Ry,Rz) as the rotation angle of the object, the rotation of the object coordinate system is performed in the order of Ry→Rx→Rz as shown in FigureC.


Object coordinate system
① Ry °
③ Rz °
② Rx °
Y
y
z
x
Z
X
5
3
9
World coordinate system
Position・rotation of object, in specifying
(Tran_X, Tran_Y,Tran_Z) = (9,5,3)
(Rot_X, Rot_Y, Rot_Z)=(Rx,Ry,Rz)。

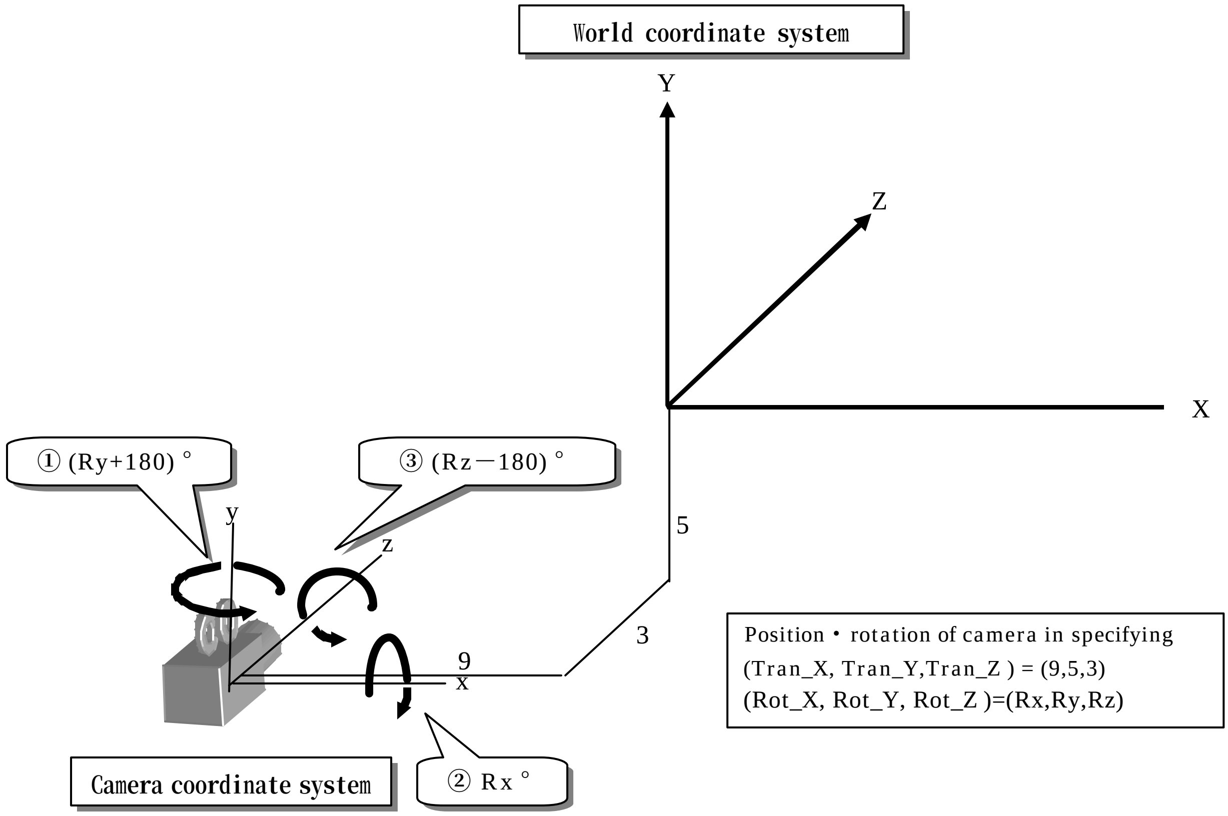
FigureC Position・rotation of object

Appendix-D Change of the motion camera

It explains the position and rotation of a camera in case of adding and altering the camera motion using 3DGL scene drawing API.

As shown in FigureD, if it specifies (Tran_X, Tran_Y, Tran_Z)=(9,5,3) as the position coordinate of a camera, it put in position the origin of the camera coordinate system for the coordinate (-9,-5,-3) of the world coordinate system.

Also, if it specifies (Rot_X, Rot_Y, Rot_Z)=(Rx,Ry,Rz) as the rotation angle of a camera, the rotation of the camera coordinate system is performed in order of (Ry+180)→Rx→(Rz−180) as shown in FigureD.


World coordinate system
Y
Z
X
① (Ry+180) °
③ (Rz−180) °
y
z
5
3
9
x
Camera coordinate system
② Rx °
Position・rotation of camera in specifying
(Tran_X, Tran_Y,Tran_Z) = (9,5,3)
(Rot_X, Rot_Y, Rot_Z)=(Rx,Ry,Rz)

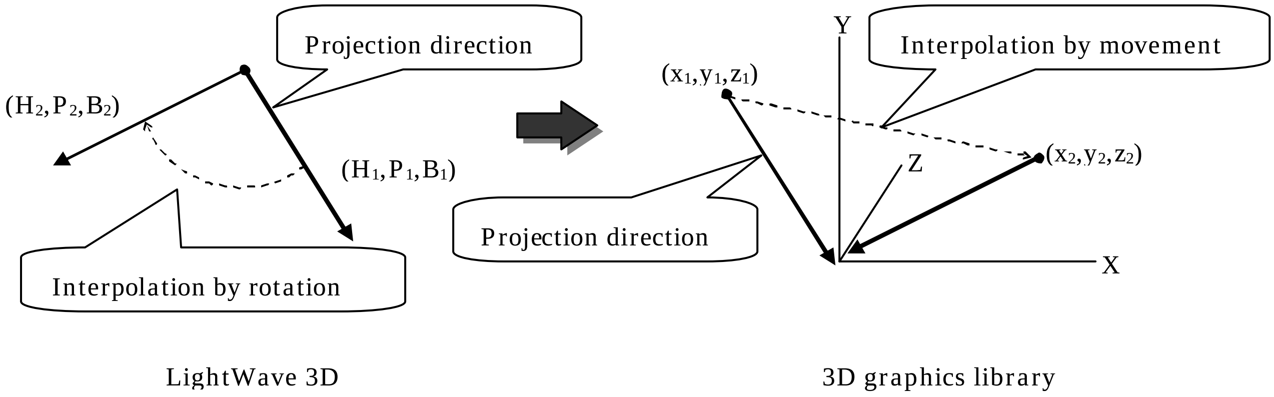
FigureD Position・rotation of camera

Appendix-E Change of the light motion

■Parallel light(Distant)

The parallel light is valid only the projection angle (projection direction). In 3D graphics library, as it expresses the projection direction of the parallel light using the light position coordinate Position[3]=(x,y,z), in the converter as shown in FigureE, it configures it calculating the light position coordinate[3] of 3D graphics library from the projection angle (H,P,B) of LightWave3D. Therefore, the position coordinate Position[3] of the parallel light is different from the position coordinate of LightWave3D.

Also, the interpolating calculation of the inter-frame is that, it is the interpolating by the rotation in LightWave3D, it is the interpolating by the movement in 3D graphics library. Therefore, configure the key frame sensitively. (if it rotates from 0℃ to 180℃, it configures the key frame, 0℃, 45℃, 90℃, 135℃, 180℃.)



FigureE Transfer of motion information of parallel light

■Point light(Point)

In case of the point light, only the position coordinate is valid. In the converter, it configures the position coordinate (x,y,z) of LightWave3D in the light position coordinate Position[3] of 3D graphics library.

■Spot light(Spot)

In the spot light, the position coordinate and the projection angle is valid. In the converter, it configures the position coordinate (x,y,z) of LightWave3D for the light position coordinate Position[3]=(x,y,x) of 3D graphics library. It calculates the projection direction from the projection angle (H, P, B) of LightWave3D, and configures it on the projection direction vector Direction[3] of 3D graphics library.