

Data Conversion Utilities

© 1998 Sony Computer Entertainment Inc.

Publication date: November 1998

Sony Computer Entertainment America
919 E. Hillsdale Blvd., 2nd floor
Foster City, CA 94404

Sony Computer Entertainment Europe
Waverley House
7-12 Noel Street
London W1V 4HH, England

The *Data Conversion Utilities* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® License and Development Tools Agreements, the Licensed Publisher Agreement and/or the Licensed Developer Agreement.

The *Data Conversion Utilities* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® License and Development Tools Agreements, the Licensed Publisher Agreement and/or the Licensed Developer Agreement.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this book is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® License and Development Tools Agreements, the Licensed Publisher Agreement and/or the Licensed Developer Agreement.

Ownership of the physical property of the book is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the book, its presentation, or its contents is prohibited.

The information in the *Data Conversion Utilities* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

PlayStation and PlayStation logos are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

Table of Contents

About This Manual	vii
Changes Since Last Release	vii
Related Documentation	vii
Developer Reference Series	vii
Typographic Conventions	viii
Developer Support	viii
Chapter 1: Video Data Conversion Utilities	
Overview	1-3
mc32: PlayStation Movie Converter	1-4
MovConv: PlayStation Movie Converter	1-49
MovPack: Packs multiple streaming/audio data sets into single pack	1-67
Chapter 2: 3D Graphic Conversion Utilities	
DXF2RSD: Converts DXF to RSD	2-3
DXF2RSDW: Converts DXF to RSD (Windows version)	2-10
LABP: Converts Labp-formatted file to binary data	2-16
MESHCONV: Creates MSH file for RSD extension	2-19
MIMEFILT: Creates MIMe difference file from TMD	2-21
MIMESORT: Sorts TMD Vertex Information	2-23
MKMIME: Makes MIMe for HMD	2-25
MKTOD: Makes TOD from PRJ and HRC files	2-27
RSD2DXF: Converts RSD to DXF	2-30
RSD2HMD: Converts RSD to HMD	2-31
RSDCAT: Links multiple RSD files to create new RSD file set	2-33
RSDFORM: Deforms or moves RSD data	2-35
RSDLINK: Converts RSD to TMD	2-39
RSDUP: Converts old RSD file to HMD-compatible RSD file	2-45
TMD2PMD: Converts TMD to PMD	2-47
TMDSORT/INFO: Sorts and displays TMD files	2-49
XHMD: Converts an HMD data file to labp format	2-50
Chapter 3: 2D Graphics Conversion Utilities	
BMP2TIM: Converts BMP to TIM	3-3
PICT2TIM: Converts PICT to TIM	3-4
RGB2TIM: Converts RGB to TIM	3-5
TIMTOOL	3-6
TIM2BMP: Converts TIM to BMP	3-10
TIMPOS: Changes VRAM in TIM	3-11
TIMUTIL: Allows multiple TIM, BMP, PICT and RGB conversions	3-12
TIMEXP: Plug-in for use with Photoshop 2.5 and 3.0	3-22
TIMexpE: Plug-in for use with Photoshop 4.0, Windows or Artist Board	3-25
TIMfmtE: Plug-in for use with Photoshop 4.0, Windows 95 or NT	3-32
TIMFMT/TIMformatE: Plug-ins for use with Photoshop 2.5 and 3.0	3-37
MGAME.ADDIN: Plug-in for Alias Wavefront	3-42
TOD_I.KXP: Plug-in for 3D Studio	3-57
Plug-ins for LightWave 3D 4.0	3-64
Chapter 4: Sound Conversion Utilities	
SMF2SEQ: Creates Sequence Data (Mac version)	4-3
SMF2SEQ: Creates Sequence Data (PC version)	4-5
AIFF2VAG: Creates Waveform Data (Mac version)	4-6
AIFF2VAG: Creates Waveform Data (PC version)	4-8
MKVAB: Makes VAB	4-9
SEQ2SEP: Converts SEQ to SEP	4-11

List of Figures

Figure 1-1: Main Window	1-6
Figure 1-2: Conversion Window	1-7
Figure 1-3: Video+Sound Window	1-8
Figure 1-4: Multi Channel Window	1-10
Figure 1-5: MDEC Parameters Window	1-12
Figure 1-6 : MDEC Parameter - Q Matrix Window	1-14
Figure 1-7: XA Parameters Window	1-14
Figure 1-8: Script Window	1-15
Figure 1-9: Video Window	1-16
Figure 1-10: Sound Window	1-17
Figure 1-11: General Setting Dialog Box	1-17
Figure 1-12: Video Data Creation Procedure	1-18
Figure 1-13: Sound Data Creation Procedure	1-21
Figure 1-14: Ordinary Streaming	1-22
Figure 1-15: Multi-channel Streaming	1-22
Figure 1-16: Efficient use of CD-ROM storage with multiple channels	1-23
Figure 1-17: Procedure for creating multi-channel data	1-23
Figure 1-18: MovConv Main Window	1-50
Figure 1-19: Preview Dialog Box	1-51
Figure 1-20: MDEC Parameters Dialog Box	1-51
Figure 1-21: Preview MDEC Image Dialog Box	1-53
Figure 1-22: Q Matrix Dialog Box	1-54
Figure 1-23: XA Audio Parameters Dialog Box	1-54
Figure 1-24: AVI Parameters Dialog Box	1-55
Figure 1-25: Interleave Video with Audio Dialog Box	1-55
Figure 1-26: Script Dialog Box	1-57
Figure 1-27: MovPack	1-67
Figure 2-1: Conversion of Coordinate Systems	2-4
Figure 2-2: Information Dialog	2-11
Figure 2-3: Information Dialog	2-13
Figure 2-4: File Open Dialog	2-14
Figure 2-5: File Save Dialog	2-14
Figure 2-6: Undefined Color Dialog	2-15
Figure 2-7: VDF File Format	2-22
Figure 3-1: Parameter Setting Window	3-14
Figure 3-2: TIM Display Dialog	3-16
Figure 3-3: File Open Dialog	3-16
Figure 3-4: File Save Dialog	3-17
Figure 3-5: Directory Select Dialog	3-18
Figure 3-6: Set Dialog	3-18
Figure 3-7: TIM Locate Dialog	3-19
Figure 3-8: VRAM Image Area	3-19
Figure 3-9: Information Area	3-20
Figure 3-10: Select List Area	3-20
Figure 3-11: Menu Bar	3-20
Figure 3-12: TIM Export Options Dialog Box	3-23
Figure 3-13: TIM Export Options Dialog Box	3-28
Figure 3-14: TIM Format Output Options Dialog Box	3-34
Figure 3-15: "Open" Dialog Box	3-38
Figure 3-16: "Open in Specified Format" Dialog Box	3-38
Figure 3-17: "Save with another Name" Dialog Box	3-39
Figure 3-18: Warning about Image Information which cannot be Saved	3-39

Figure 3-19: TIM Format Output Options dialog box	3-40
Figure 3-20: Opening the Plug-in Manager window	3-42
Figure 3-21: Search path and file name window	3-42
Figure 3-22: Plug-in Manager window	3-43
Figure 3-23: Opening the Make Game translator window	3-43
Figure 3-24: Make Game translator window	3-44
Figure 3-25: PlayStation Export window	3-45
Figure 3-26: TMD File option	3-45
Figure 3-27: Video RAM Mapper window	3-47
Figure 3-28: Screen Size Menus	3-48
Figure 3-29: Load Map File window	3-48
Figure 3-30: Textures section	3-49
Figure 3-31: Options Section	3-50
Figure 3-32: Animation Section	3-50
Figure 4-1: SMF Output in Performer	4-3
Figure 4-2: SMF2SEQ	4-4
Figure 4-3: SMF2SEQ Icon	4-4
Figure 4-4: AIFF Output in SoundDesignerII	4-6
Figure 4-5: AIFF2VAG Conversion File	4-6
Figure 4-6: AIFF2VAG Icon	4-7
Figure 4-7: ADSR rates	4-10

List of Tables

Table 1-1: Movie Tools	1-3
Table 1-2: Formats Supported by mc32	1-4
Table 1-3: Setting the Total Number of Channels for XA Sound	1-11
Table 1-4: Type of conversion as a function of input data and output format settings	1-12
Table 1-5: Setting leap sectors	1-13
Table 1-6: Script Functions	1-25
Table 1-7: Conversion Differences as a Function of Audio	1-52
Table 1-8: Leap Sector Setting	1-52
Table 1-9: List of Scripting Functions	1-59
Table 1-10: Settings for the total number of channels in XA audio data	1-68
Table 2-1: Commercial 3D modelers convertible with DXF2RSD	2-5
Table 2-2: Currently unconvertible commercial 3D modelers	2-6
Table 2-3: RSD → TMD Conversion Comparison Table	2-41
Table 3-1: Parameters specifiable in the [DefaultValue] section of TIMexp.ini	3-27
Table 3-2: Parameters specifiable in the [DefaultValue] section of TIMfmt.ini	3-33
Table 3-3: Controlling the STP bit of the color entries	3-49
Table 3-4: GetPlayStation Options	3-51
Table 4-1: Appropriate Format (SMF)	4-3
Table 4-2: VabHdr	4-9
Table 4-3: ProgAtr	4-9
Table 4-4: VagAtr	4-10

About This Manual

This manual is the latest release of documentation for PlayStation® Data Conversion Utilities as of Run-Time Library 4.4. This document is a compilation of all available file conversion information. Some sections may also exist as individual documents within the *Developer Reference Series*. The purpose of this manual is to provide a single authoritative reference to PlayStation® Data Conversion Utilities.

Changes Since Last Release

- PC versions of two sound converters: smf2seq and aiff2vag have been added to the existing Mac version descriptions.
- The release history and revision number information in XHMD and MKMIME have been updated.
- A path change was made in LABP to match changes in SDevTC.

Related Documentation

The following volumes in the *Developer Reference Series* also contain file format information:

- File Formats
- 3D Graphics Tool
- Sound Artist Tool

Developer Reference Series

This manual is part of the *Developer Reference Series*, a series of technical reference volumes covering all aspects of PlayStation development. The complete series is listed below:

Manual	Description
PlayStation Hardware	Describes the PlayStation hardware architecture and overviews its subsystems.
PlayStation Operating System	Describes the PlayStation operating system and related programming fundamentals.
Run-Time Library Overview	Describes the structure and purpose of the run-time libraries provided for PlayStation software development.
Run-Time Library Reference	Defines all available PlayStation run-time library functions, macros and structures.
Inline Programming Reference	Describes in-line programming using DMPSX, GTE inline macro and GTE register information.
SDevTC Development Environment	Describes the SDevTC (formerly "Psy-Q") Development Environment for PlayStation software development.
3D Graphics Tools	Describes how to use the PlayStation 3D Graphics Tools, including the animation and material editors.
Sprite Editor	Describes the Sprite Editor tool for creating sprite data and background picture components.
Sound Artist Tool	Provides installation and operation instructions for the DTL-H800 Sound Artist Board and explains how to use the Sound Artist Tool software.
File Formats	Describes all native PlayStation data formats.

Data Conversion Utilities	Describes all available PlayStation data conversion utilities, including both stand-alone and plug-in programs.
CD Emulator	Provides installation and operation instructions for the CD Emulator subsystem and related software.
CD-ROM Generator	Describes how to use the CD-ROM Generator software to write CD-R discs.
Performance Analyzer User Guide	Provides general instructions for using the Performance Analyzer software.
Performance Analyzer Technical Reference	Describes how to measure software performance and interpret the results using the Performance Analyser.
DTL-H2000 Installation and Operation	Provides installation and operation instructions for the DTL-H2000 Development System.
DTL-H2500/2700 Installation and Operation	Provides installation and operation instructions for the DTL-H2500/H2700 Development Systems.

Typographic Conventions

Certain Typographic Conventions are used through out this manual to clarify the meaning of the text. The following conventions apply to all narrative text except for structure and function descriptions:

<i>Convention</i>	<i>Meaning</i>
<code>courier</code>	Indicates literal program code.
Bold	Indicates a document, chapter or section title.

The following conventions apply within structure and function descriptions only:

<i>Convention</i>	<i>Meaning</i>
Medium Bold	Denotes structure or function types and names.
<i>Italic</i>	Denotes function arguments and structure members.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
<i>In North America</i> Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd., 2nd floor Foster City, CA 94404 Tel: (650) 655-8000	<i>In North America</i> E-mail: DevTech_Support@playstation.sony.com Web: http://www.scea.sony.com/dev Developer Support Hotline: (650) 655-8181 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

SCEE developer support is available to licensees in Europe only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
<i>In Europe</i> Attn: Production Coordinator Sony Computer Entertainment Europe Waverley House 7-12 Noel Street London W1V 4HH Tel: +44 (0) 171 447 1600	<i>In Europe</i> E-mail: dev_support@playstation.co.uk Web: https://www-s.playstation.co.uk Developer Support Hotline: +44 (0) 171 447 1680 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT or BST/BDT)

Chapter 1:

Video Data Conversion Utilities

Overview

The PlayStation can successively read and process data from the CD-ROM. This operation is known as “streaming”, and is commonly used to play back video and sound. For processing video, the PlayStation uses a frame-based compressed video format that is JPEG-compatible. During playback, each compressed frame is expanded using the MDEC, the PlayStation's image expansion chip, and the expanded frame is presented on the display. In a similar manner, sound processing is performed with CD-ROM XA compression. During playback, sound is reproduced using the PlayStation's CD-ROM XA decoder.

Utilities called *movie tools* compress and format video and sound data into formats suitable for use by the PlayStation's hardware and libraries. Conventional movie tools such as movconv and movpack are available for Windows 3.1 and Windows 95. The new movie converter, mc32, which runs under Windows 95 and Windows NT, integrates these functions and provides much faster image compression.

The next section describes the functionality and use of mc32. The previous movie tools, movconv and movpack, are described in subsequent sections.

Table 1-1: Movie Tools

Program name	Functions, environment
mc32.exe	Compression of video and sound data Format conversion Multichannel interleave (Windows 95/NT)
Movconv.exe	Compression of video and sound data Format conversion (Windows 3.1/95)
Movpack.exe	Multichannel interleave (Windows 3.1/95)

mc32: PlayStation Movie Converter

Overview

Features

mc32 provides video compression and sound compression, format conversion, multiple channel creation, video/sound synthesis, script execution, and other functions.

During video compression, various parameters such as the frame rate must be specified. With mc32, these parameters can be selected while simultaneously verifying the quality of the compressed image. mc32 also supports a simple scripting language which allows batch conversion of multiple data items.

Functions

mc32 performs the following five functions:

- Video conversion
- Sound conversion
- Video/sound synthesis
- Creation of multiple channels of data
- Script execution

Supported Formats

mc32 supports the following input and output formats:

Table 1-2: Formats Supported by mc32

Format name	Type	Description	Input	Output
AVI	Video	Windows video	X	X
TIM sequence	Video (still-image sequence)	PlayStation still-image sequence	X	
RGB sequence	Video (still-image sequence)	Uncompressed RGB still-image sequence	X	X
YUV sequence	Video (still-image sequence)	Uncompressed YUV still-image sequence	X	
STR(MDEC)	Video	MDEC-compressed video	X	X
BS sequence	Video (still-image sequence)	MDEC-compressed still-image sequence	X	X
WAV	Sound	Windows sound	X	
PCM	Sound	Raw sound	X	
XA	Sound	CD-ROM – XA compressed sound		X

AVI

AVI is the Windows' video format. Although several types of AVI compression are available, it is advisable to use uncompressed AVI whenever possible. The use of AVI compression may result in deterioration of the quality of MDEC-encoded images.

TIM Sequence

TIM is the PlayStation's still-image format. mc32 interprets a sequence of TIM files as a series of images. TIM files are sequenced according to the following conventions.

```

...
...
xx0017.TIM
xx0018.TIM
xx0019.TIM
xx0020.TIM
...
...

```

(xx is any character string, and there need not be four numeric digits.)

To specify a TIM sequence, specify the name of the file corresponding to the first frame. Once the first frame is specified, the files corresponding to this and subsequent frames are retrieved automatically from the same directory. The collection of frames is handled as a series of images.

RGB Sequence

RGB is a 24-bit direct color format. Each pixel in RGB is represented by 8 bits of red, 8 bits of green and 8 bits of blue. The pixels are ordered such that the first pixel is in the upper-left corner and the last pixel is in the lower-right corner.

Width and height are not represented in the RGB format. When an RGB file is specified, mc32 will request that the width and height also be specified. The file naming conventions and specification method for the RGB format are the same as those for the TIM sequence.

YUV Sequence

The YUV (422) format orders the luminance (Y) and two color difference components (Cb, Cr) as follows:

```

Cb12, Y1, Cr12, Y2, Cb34, Y3, Cr34, Y4 ....

```

Each component includes an 8-bit gradation. In addition, the same color-difference data is shared by two adjacent pixels, so the horizontal resolution of the color difference component is one-half the horizontal resolution of the luminance component.

Width and height are not represented in the YUV format. When a YUV file is specified, mc32 will request that the width and height also be specified. The file naming conventions and specification method for the YUV format are the same as those for the TIM sequence.

STR(MDEC)

STR(MDEC) is an STR format with each frame MDEC-compressed. The format of each STR(MDEC) frame is the same as that of the BS format.

BS Sequence

BS is an MDEC-compressed still image format. Width and height are not represented in the BS format. The file naming conventions and specification method for the BS format are identical to those for the TIM sequence.

WAV

WAV is the Windows' sound format. In mc32, WAV is equivalent to a PCM format with sampling frequencies of 44 KHz / 22 KHz / 11 KHz and a sample length of 16 bits.

PCM

PCM is an uncompressed sound format. It can be used with data having a 16-bit sample length.

XA

XA is a format for ADPCM compression in accordance with the CD-ROM XA standard. The sampling frequencies of XA are 37.8 KHz and 19.8 KHz, corresponding to stereo and monaural, respectively.

mc32 Operation

This section describes the basic operation of mc32. For GUI-based operation, the use of each window and dialog box is described in detail below.

Main Window

Figure 1-1: Main Window

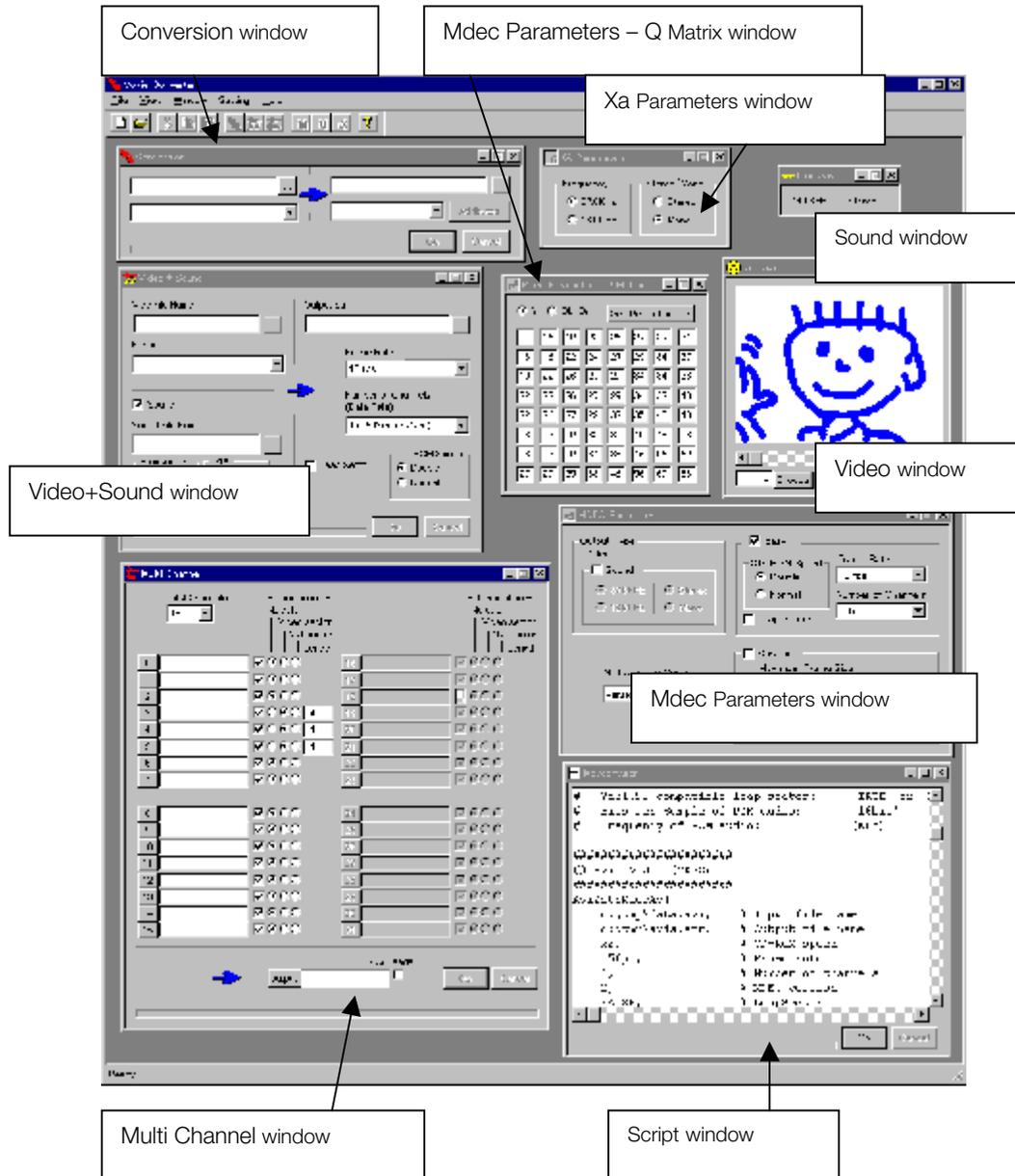
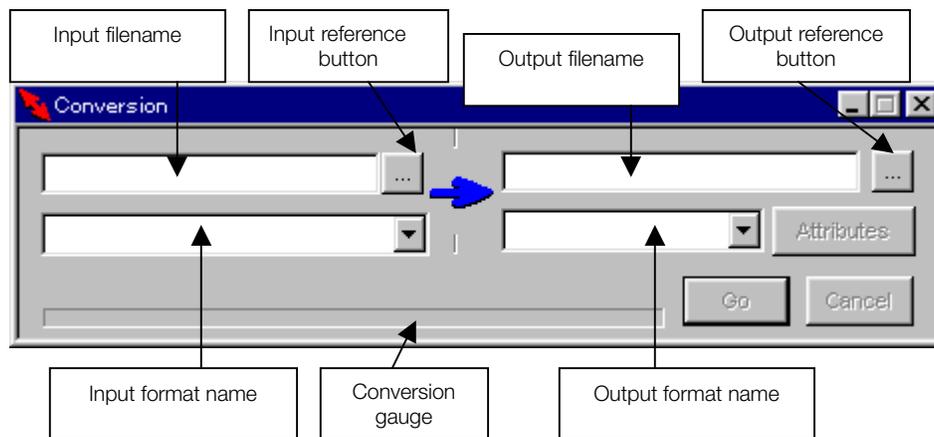


Figure 1-13 shows the Main window of mc32, the initial window that is displayed after starting mc32. From the Main window, the following child windows can be opened from the View menu, from the File/Open menu, or by using the toolbar:

- Conversion window
- Video+Sound window
- Multi Channel window
- Mdec Parameters window
- Mdec Parameter - Q Matrix window
- Xa Parameters window
- Script window
- Video window
- Sound window

Conversion Window

Figure 1-2: Conversion Window



The Conversion window is used to perform Video MDEC compression and sound CD-ROM XA compression. MDEC expansion can also be performed by specifying a compressed image as input. The Conversion window can be opened from the View/Conversion menu or by using the Conversion button on the toolbar. The Conversion window is also displayed in the Main window after mc32 is started.

- **Specifying the Input Filename**
The input filename is specified by clicking the input reference button ([...] button on the left) and opening the File Open dialog box. The input filename can also be entered directly in the Filename text box on the left. When a file is specified with the input reference button, the specified file is displayed in the Video window. This allows the contents of the input file to be verified prior to conversion.
- **Specifying the Input Format**
The input format is inferred from the input filename extension and is selected automatically. The input format can also be selected from the Format list on the left side of the window.
- **Specifying the Output Filename**
The output filename is inferred from the input filename and the output format, and is automatically displayed. The output filename can be changed by clicking the output reference button ([...] button on the right). This will open the Save File dialog box and allow the output filename to be selected. The output filename can also be entered directly in the Filename text box on the right side of the window.
- **Specifying the Output Format**
The output format can be selected from the Format list on the right side of the window.

- **Setting the Output Parameters**

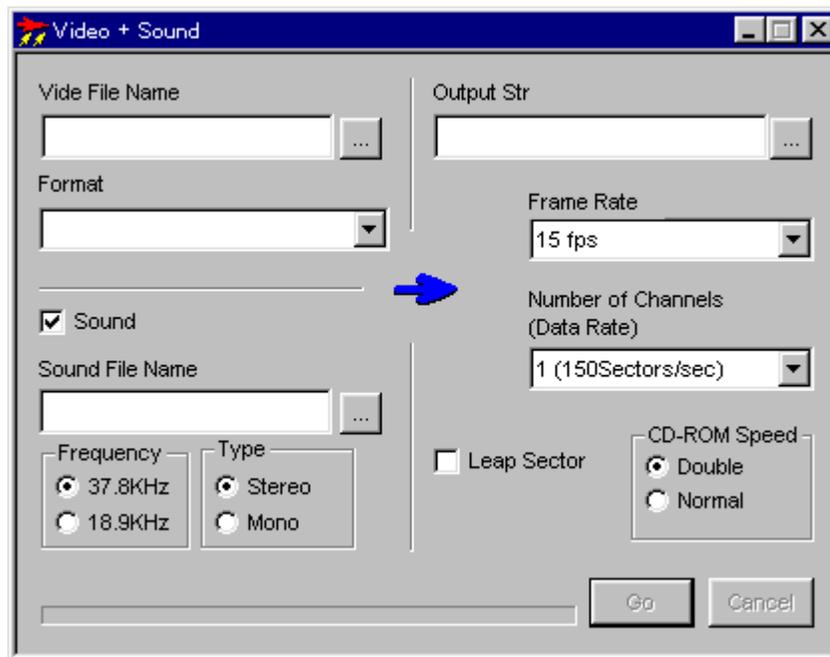
After selecting the output format, the output parameters can be selected by clicking the Attributes button. This will open the parameter window for the selected output format. There are two parameter windows: the MDEC Parameters window for MDEC compression and the XA Parameters window for sound compression. These windows can also be opened from the View menu or by using the corresponding button on the toolbar.

- **Conversion**

Once all settings have been made, the conversion operation can be started by clicking the Go button. The conversion parameter values set in the parameters window are used during conversion, however, changes in their values will not be reflected in the conversion. Progress is displayed by a gauge during conversion. The conversion operation can be cancelled by clicking the Cancel button.

Video+Sound Window

Figure 1-3: Video+Sound Window



The Video+Sound window is used to perform video/sound synthesis. It can be opened from the View/Video+Sound menu or by using the Video+Sound button on the toolbar.

When the input data contains both video and sound, and is converted to STR(MDEC) with sound in the Conversion window, video and audio synthesis is automatically performed. In this case, it is not necessary to explicitly perform the synthesis in the Video+Sound window. Synthesis is only required when automatic conversion is not performed.

- **Specifying the Video Filename**

The video filename can be specified by clicking the [...] button next to Video File Name. This will open the File Open dialog box and allow the video file to be selected. The video filename can also be entered directly in the File Name text box.

- **Specifying the Video Format**

The video format can be specified by selecting it from the Format list.

- **Specifying Sound / No Sound**

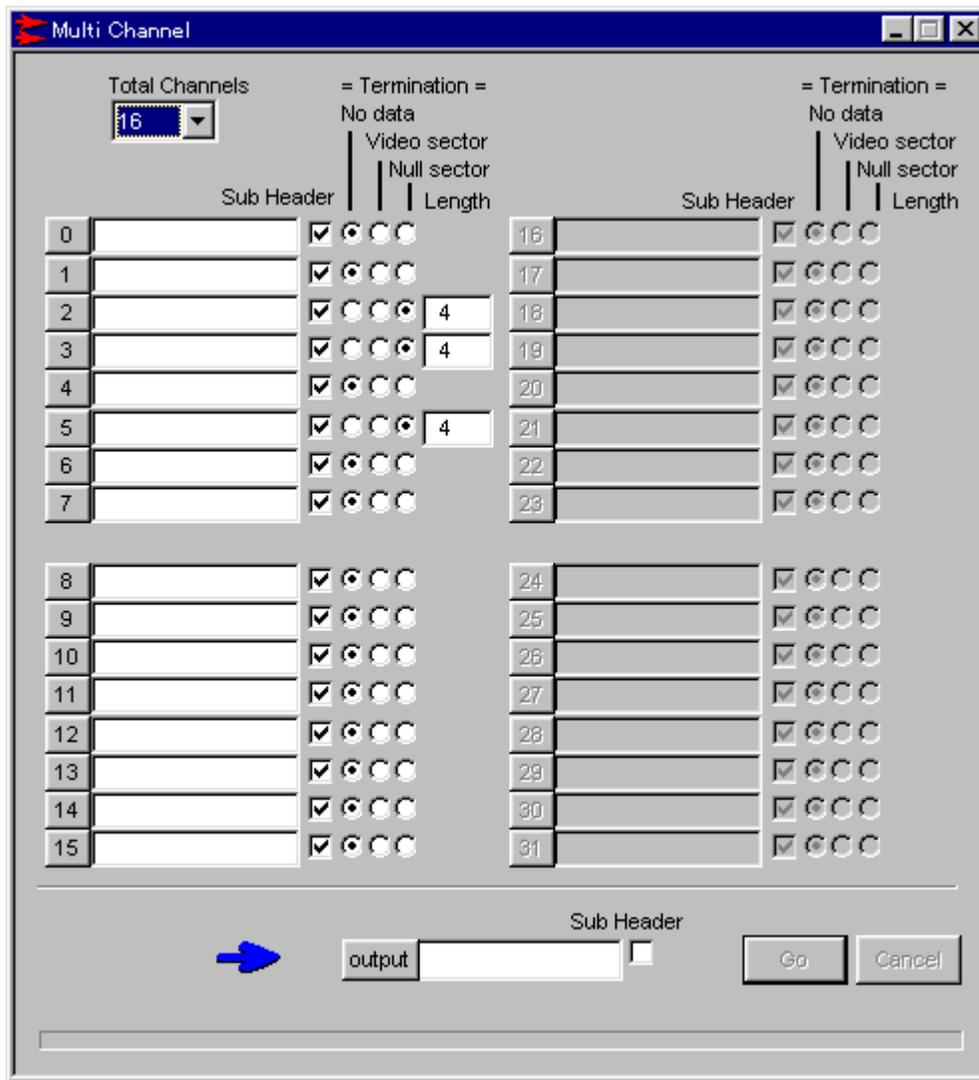
Sound / No Sound mode can be specified by checking/unchecking the Sound check box. If video/sound synthesis is required, the Sound check box should be checked. If it is only necessary to convert the video data without the need for synthesis, the Sound check box should be left unchecked.

For example, when converting a BS sequence to STR(MDEC), the Sound check box should not be checked.

- **Specifying the Sound Filename**
After selecting the sound mode, the sound filename can be specified by clicking the [...] button next to Sound File Name. This will open the File Open dialog box and allow the sound filename to be selected. The sound filename can also be entered directly in the File Name text box.
- **Setting the Frequency and Stereo/Monaural Mode**
After selecting the sound mode, the frequency and stereo / monaural mode can be selected. The attributes of the specified sound file should be selected and must match those of the actual file, otherwise video/audio synthesis will not be performed correctly.
- **Setting the CD-ROM Speed**
The CD-ROM speed can be selected as Double speed or Normal speed in the CD-ROM Speed group.
- **Setting Leap Sectors**
Leap Sectors are set by checking the Leap Sector check box. See the description under the MDEC Parameters window on how to determine if leap sectors are required.
- **Selecting the Frame Rate**
The frame rate can be selected from the Frame Rate list. This value must agree with the frame rate of the video file as frame rate conversion is not performed. If the selected frame rate were different from the original frame rate, the playback time of the video would not be correct.
- **Selecting Number of Channels**
The number of channels can be selected from the Number of Channels (Data Rate) list and is used to create multichannel data. Select '1' if multichannel data is not required.
- **Specifying the Output Filename**
The output filename can be specified by clicking the [...] button of the Output Str. This will open the File Save dialog box and allow the output filename to be selected. The output filename can also be entered directly in the Output Str text box.
- **Execution**
Video/sound synthesis can be started by clicking the Go button. Progress is displayed by a gauge during synthesis. The synthesis operation can be cancelled by clicking the Cancel button.

Multi Channel Window

Figure 1-4: Multi Channel Window



The Multi Channel window can be used to create multiple channels of interleaved XA sound data and STR(MDEC) video data. The Multi Channel window can be opened from the View/Multi Channel menu or by using the Multi Channel button on the toolbar. It is not necessary to use this window when creating ordinary one-channel STR(MDEC) video. After XA sound has been created using the Conversion window, multiple channels must be created in the Multi Channel window before burning the data onto a CD-ROM.

The Multi Channel window should not be used to synthesize STR(MDEC) video and XA sound. Instead, the Video+Sound window should be used for this operation.

- **Setting the Total Number of Channels**

The total number of channels can be set from the Total Channels list. If the multichanneled data is XA sound, set the total number of channels according to the values shown in the following table.

Table 1-3: Setting the Total Number of Channels for XA Sound

Playback speed	Sampling frequency	Stereo/monaural	Total number of channels
Double speed	37.8KHz	Stereo	8
Double speed	37.8KHz	Monaural	16
Double speed	18.9KHz	Stereo	16
Double speed	18.9KHz	Monaural	32
Standard speed	37.8KHz	Stereo	4
Standard speed	37.8KHz	Monaural	8
Standard speed	18.9KHz	Stereo	8
Standard speed	18.9KHz	Monaural	16

If XA sound is converted in the Conversion window then burned directly onto a CD-ROM, it will not play back correctly. To ensure correct playback, it is necessary to interleave the sound data according to the number of channels shown in the table.

- **Specifying the Input Filename**

An input channel filename must be specified for each channel. The input filenames can be specified by clicking each of the channel's buttons. This will open the File Open dialog box and allow the input filename to be specified. The input filename can also be entered directly in the text box.

If the number of files to be interleaved is less than the total number of channels, the extra channels may be left blank. Any unspecified channels will be treated as empty channels. For video data, the input file should be specified as STR(MDEC). For sound data, the input file should be specified as XA.

- **Setting the Input Subheader Flag**

Set the subheader flag for those channels in which an input file was specified. The Sub Header check box should be checked if the input has either sound data or streaming data with sound. The Sub Header check box should be left unchecked if the input is video-only streaming data.

- **Setting the Data Terminator**

The data terminator that is appended to the end of each channel can be set by selecting the appropriate radio button under Termination. Normally, No data is selected. When Video sector is selected, an empty video sector is appended to the end of the channel data. Empty video sectors can be used for creating multiple sound channels. If an empty video sector is appended to the end of sound data, reading the video sector during playback can be used to identify the end of sound data.

Null sector can be selected to intentionally insert a space at the end of channel data. When Video sector or Null sector is specified for the data terminator, the length is specified in sectors.

- **Specifying the Output Filename**

The output filename can be specified by clicking the output button. This will open the Save File dialog box and allow the output filename to be selected. The output filename can also be specified by entering the filename directly in the text box.

- **Setting the Output Data Subheader Flag**

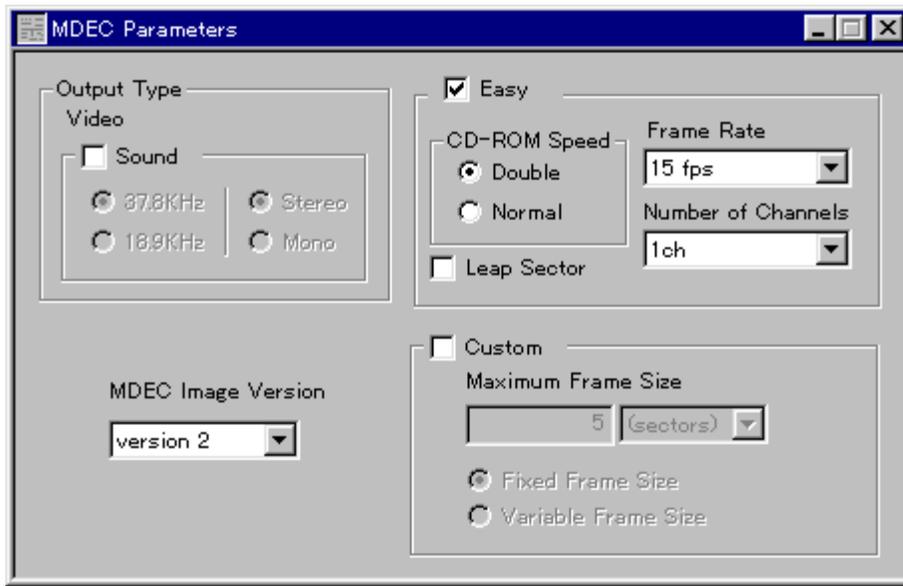
The Output Data Sub Header check box should be checked if the output data is either streaming data with sound or sound data. Leave the Sub Header check box unchecked if the output is video-only streaming data.

- **Interleaving**

Interleaving can be started by clicking the Go button. Progress is displayed by a gauge during the operation. Interleaving can be cancelled by clicking the Cancel button.

MDEC Parameters Window

Figure 1-5: MDEC Parameters Window



The MDEC Parameters window is used to set compression parameters for MDEC compression. The window can be opened from the View/MDEC Parameters menu, or by using the MDEC Parameters button on the toolbar or the Attributes button in the Conversion window. The values which are specified in the MDEC Parameters window are used as conversion parameters at the start of MDEC compression in the Conversion window and during MDEC compression in the Video window. If values are changed once conversion has started, the updated values will not be reflected in the converted data. In addition to the MDEC compression parameters shown above, other parameters which can be set include the quantization matrix which is specified in the MDEC Parameter - Q Matrix window.

- **Selecting the Output Type**

The output type is specified using the Sound check box in the Output Type group. Check the Sound check box if the output data should contain sound, and leave the box unchecked if sound should not be included in the output. The output type indicates whether the output data should contain sound, and does not depend on whether the input data contains sound. The following table shows the type of conversion performed as a function of input data and output format settings.

Table 1-4: Type of conversion as a function of input data and output format settings

Input data	Output settings	Operation
With sound	With sound	Video conversion, sound conversion, video/sound synthesis
With sound	Without sound	Video conversion only
Without sound	With sound	Video conversion only. The converted video data size does not include the sound component. The video and sound are converted separately, and this setting is used later when performing video/sound synthesis.
Without sound	Without sound	Video conversion only

- **Selecting the Frequency and Stereo/Monaural Mode**

When the Sound checkbox is checked, the output sampling frequency and stereo/monaural mode can be selected using the radio buttons in the Sound group.

- **Selecting Easy Settings or Custom Settings**

Easy Settings and Custom Settings are two methods for setting parameters in the MDEC Parameters window.

When Easy Settings is selected, the size of each frame after compression is determined automatically from parameters such as the CD-ROM playback speed, frame rate and number of channels. Easy Settings should be used when the size of the compressed image does not need to be controlled.

When Custom Settings is selected, the maximum amount of storage used by the compressed image can be specified. This is useful when compressing still images or individual frames of video that need to be smaller than a certain size. When the output data contains sound, only Easy Settings can be used.

Check the Easy check box to select Easy Settings. Check the Custom check box to select Custom Settings.

- **Selecting the CD-ROM Playback Speed**

If Easy Settings is selected, the radio buttons in the CD-ROM Speed group can be used to select double speed or standard speed.

- **Selecting the Frame Rate**

If Easy Settings is selected, an appropriate frame rate can be selected from the Frame Rate list.

- **Selecting the Number of Channels**

If Easy Settings is selected, an appropriate number of channels can be selected from the Number of Channels list.

- **Setting a Leap Sector**

If Easy Settings is selected, a leap sector can be specified. Leap sectors are used to correct deviations that take place due to the difference between the 59.94fps frame rate used in video devices and the 60fps frame rate expected by the PlayStation for streaming from a CD-ROM. When leap sectors are enabled, an extra sector is inserted at regular intervals. The use of the leap sector setting is shown in the table below.

Table 1-5: Setting leap sectors

Capture method	Leap sector
Capture from videotape (59.94fps material)	on
Material created with a 60-fps unit frame rate in CG rendering, etc. (60-fps material)	off

- **Setting the Maximum Frame Size**

If Custom Settings is selected, the maximum frame size for single compressed frames can be specified in Maximum Frame Size. Units can be specified in either sectors or bytes. The relationship between the two is as follows:

$$1 \text{ sector} = 2016 \text{ bytes}$$

- **Setting the Frame Integration Method**

If Custom Settings is selected, the method used to integrate frames can be specified in the Custom group box.

The setting for Maximum Frame Size specifies an upper bound on the compressed data size, so the actual compressed size may be less. If Fixed Frame Size is specified, dummy data will be inserted after the compressed data so that the resultant frame size will be at the maximum value.

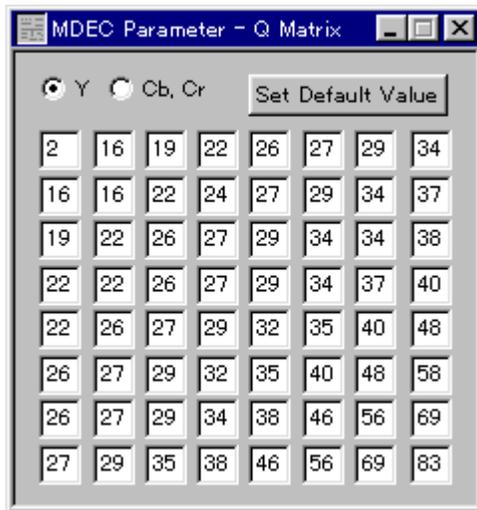
Dummy data will not be inserted if Variable Frame Size is specified.

- **Selecting the Compression Format Version**

The version of the compression format is selected using the MDEC Image Version list. Although the image quality of version 3 is slightly better than that of version 2, decoding is slower, so version 2 is normally used.

MDEC Parameter - Q Matrix Window

Figure 1-6 : MDEC Parameter - Q Matrix Window

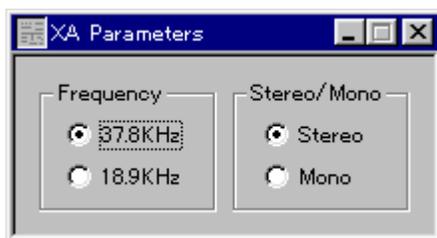


The MDEC Parameter - Q Matrix window is used to change the MDEC compression quantization matrices. The window can be opened either from the View/MDEC Parameter - Q Matrix menu or by using the MDEC Parameter - Q Matrix button on the toolbar. The respective values for the quantization matrices can be set in the luminance block and the color-difference block. The quality of a specific image can be optimized by changing the quantization matrix, but the default values normally are satisfactory. If the quantization matrix is changed, the same matrix must be used for decoding. libpress provides a function for performing this operation. The default quantization matrix is the same as the standard MPEG quantization matrix. For more information on the quantization matrix, please refer to the documentation for MPEG, JPEG, and other related information.

- **Selecting the Luminance or Color-Difference Block**
Radio button Y is used to select the quantization matrix for the luminance block and radio button Cb, Cr is used to select the quantization matrix for the color-difference block.
- **Changing the Quantization Matrix**
The quantization matrix can be changed by directly editing the matrix values.
- **Setting the Default Quantization Matrix**
The default quantization matrix can be set by clicking the Set Default Value button.

XA Parameters Window

Figure 1-7: XA Parameters Window

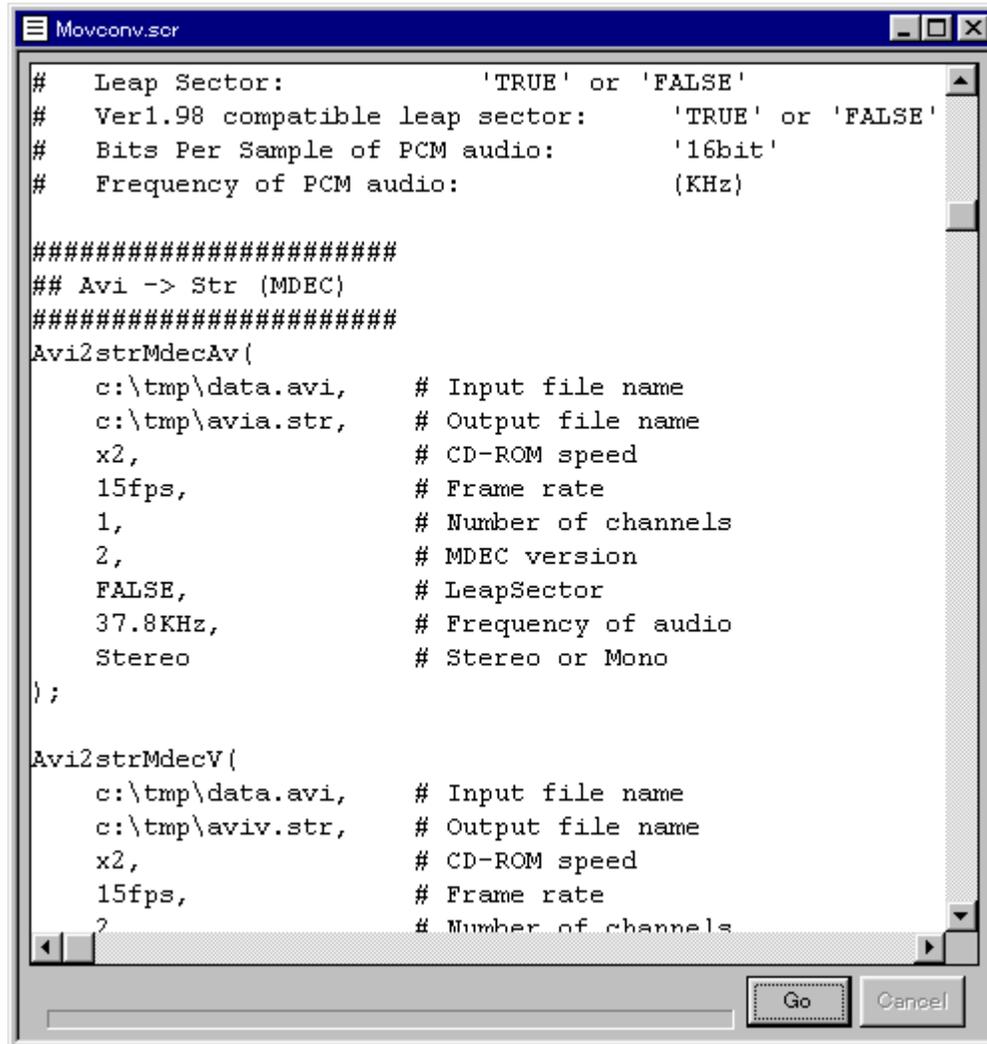


The XA Parameters window is used to set the XA compression parameters. The window can be opened either from the View/XA Parameters menu or by using the XA Parameters button on the toolbar. During XA compression, the values specified in the XA Parameters window are used as the conversion parameters in the Conversion window. The frequency is automatically converted.

- **Selecting the Frequency**
The sampling frequency for XA data can be selected using the radio buttons in the Frequency group.
- **Selecting Stereo or Monaural**
Stereo or monaural mode can be selected using the radio buttons in the Stereo/Mono group.

Script Window

Figure 1-8: Script Window



```

Movconv.scr
# Leap Sector:          'TRUE' or 'FALSE'
# Ver1.98 compatible leap sector:  'TRUE' or 'FALSE'
# Bits Per Sample of PCM audio:    '16bit'
# Frequency of PCM audio:          (KHz)

#####
## Avi -> Str (MDEC)
#####
Avi2strMdecAv(
    c:\tmp\data.avi,      # Input file name
    c:\tmp\avia.str,     # Output file name
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    1,                   # Number of channels
    2,                   # MDEC version
    FALSE,               # LeapSector
    37.8KHz,             # Frequency of audio
    Stereo                # Stereo or Mono
);

Avi2strMdecV(
    c:\tmp\data.avi,      # Input file name
    c:\tmp\aviv.str,     # Output file name
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    ?                    # Number of channels

```

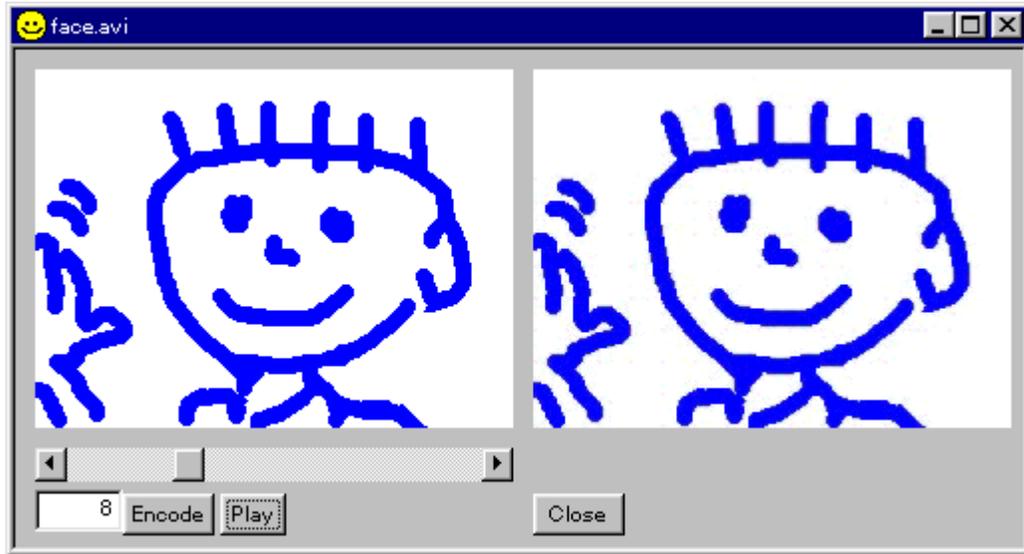
Go Cancel

The Script window is used to initiate batch conversions from a script file. The Script window can be opened from either the File/New menu or the File/Open menu. If the Script window is opened from the File/New menu, an empty script will be displayed. If the Script window is opened from the File/Open menu, the script which is opened will be displayed.

- **Editing a Script**
A script which is displayed can be edited directly. Standard editing operations (i.e., Copy, Paste, Cut, Undo) can be performed either from the Edit menu, by using the corresponding buttons on the toolbar, or through shortcuts. Large scripts cannot be edited in the Script window. A text editor should be used to edit a large script outside of mc32.
- **Running a Script**
A script can be started by clicking the Go button. Progress is displayed by a gauge during execution. The script can be cancelled by clicking the Cancel button.

Video Window

Figure 1-9: Video Window



The Video window is used to verify an input image or to check the quality of an MDEC-compressed image on the PC. The Video window can be opened from the File/Open menu or by using the input reference button in the Conversion window, after reading a video or still-image file. Because the colorimetry of a PC monitor may be different from the colorimetry of the monitor attached to the PlayStation, the final image quality should be verified using actual equipment.

- **Playing a Video**

A video can be played back from the current frame to the last frame by pressing the Play button. The desired playback speed is set by the Video Play Back Frame Rate in the General Setting dialog box. Note that the actual playback speed will depend on the current PC load and whether or not other videos are being simultaneously played back. If processing is too slow, the frame rate should be reduced to avoid dropping frames. While a video is playing, the Play button, which will appear as the Stop button, can be used to terminate playback. The quality of a compressed image can be roughly determined by playing back the STR(MDEC) data.

- **Displaying an Arbitrary Frame**

A frame can be displayed by directly entering its frame number or by operating the slider.

- **Checking the Image Quality after MDEC Compression**

The quality of an image after MDEC compression can be checked by clicking the Encode button. This will display the result of compressing/expanding the currently displayed frame. Compression is controlled by the compression parameter values in the MDEC Parameters window and the MDEC Parameters - Q Matrix window.

Sound Window

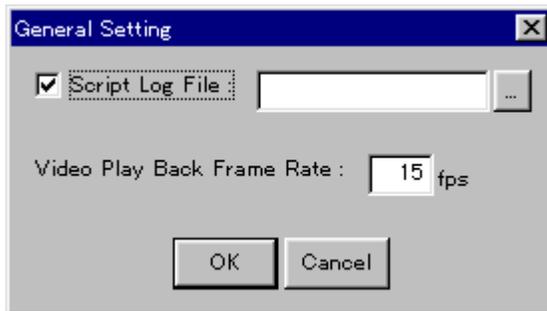
Figure 1-10: Sound Window



The Sound window displays the frequency and stereo/monaural settings of the input file. It does not accept any input data. The Sound window can be opened from the File/Open menu or by using the input reference button in the Conversion window after a sound file is read.

General Setting Dialog Box

Figure 1-11: General Setting Dialog Box



The General Setting dialog box is used for making general-purpose settings in mc32. The General Setting dialog box is opened from the Setting/General Setting menu.

- Setting the Script Log Filename**
 During execution of a script, warning messages normally appear in a dialog box. These messages can be written to an output file by checking the Script Log File check box and specifying the log filename. The log file is particularly useful when performing a large number of conversions using a script.
- Setting the Video Playback Frame Rate**
 The frame rate of video playback can be set in the Video Play Back Frame Rate input field in the General Setting dialog box. Note that the actual playback speed will depend on the current PC load and whether or not other videos are being simultaneously played back. If processing is too slow, the frame rate should be reduced to avoid dropping frames.

Data Conversion without Using a GUI

mc32 can perform its processing without displaying a window.

Use the command:

```
> mc32 -s script_file
```

to execute mc32 on the DOS command line window without a GUI.

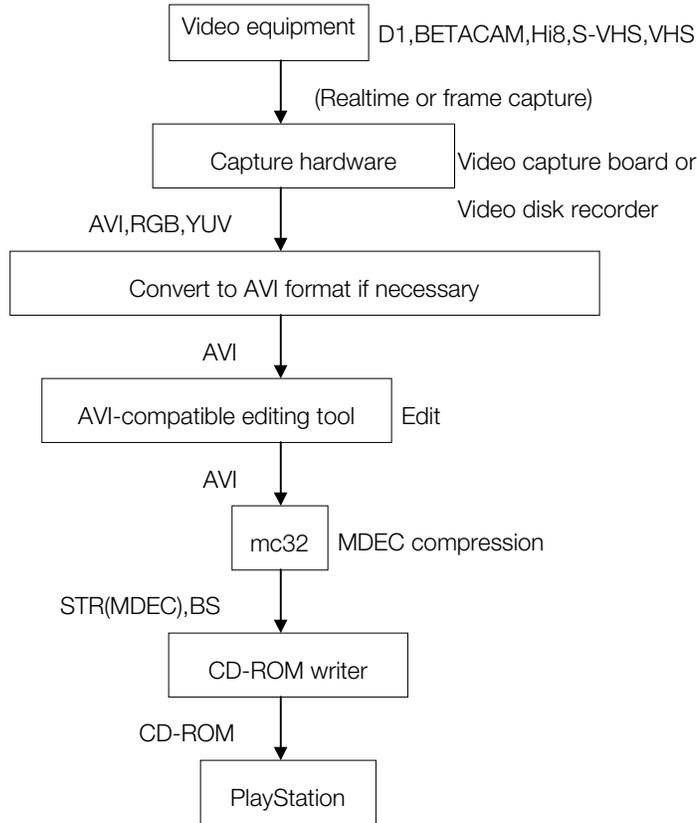
The argument "script_file" indicates the specified script. Using a script file allows mc32 to be invoked from a batch file or other external script.

Creating Video Data

Video Creation Procedure

The most basic function that mc32 provides is the creation of an MDEC video. The following figure shows the processing steps from video capture through MDEC video creation.

Figure 1-12: Video Data Creation Procedure



First, the video recorded on video media (e.g., D1, Betacam, Hi8, S-VHS, VHS) is captured by means of video capture equipment. Examples of capture equipment include videodisk recorders and PC video capture boards. The capture format depends on the specific capture equipment that is used, but typical formats include RGB, YUV, AVI, etc. For additional information on these formats, please refer to the section on supported formats.

If the capture format is AVI, the images can be edited as is with an editing tool on the PC. If the capture format is RGB or YUV, the images should first be converted to AVI using mc32, then edited with the PC's editing tool. For other capture formats, the images should first be converted to AVI using an appropriate conversion tool, then subsequently edited using an editing tool. If the images are pre-edited before data capture, the PC editing step may be eliminated.

Next, the post-edited AVI data is converted to MDEC video using mc32. Finally, the compressed video is burned onto a CD-ROM using a CD-ROM writer. The conversion of AVI data to MDEC video using mc32 will be explained in more detail in the next subsection. For information on creating a CD-ROM using a CD-ROM writer, refer to the subsection entitled "Burning Streaming Data onto a CD-ROM."

mc32 does not provide hardware or software for video capture, so standard capture equipment should be used. In addition, mc32 does not provide a video editing tool, so a separate editing tool should be used.

In the following section, window functions are only briefly discussed. For more detailed information, refer to the section entitled "mc32 Operation."

Creating Video Data Without Sound

MDEC video is broadly divided into two types: video without sound and video with sound. MDEC video uses the sector image of the CD-ROM as its basic format. However, sector size will vary depending on whether the data type is video with sound or video without sound. For video with sound, 1 sector = 2336 bytes, and image data is interleaved with sound data at a fixed rate. For video without sound, 1 sector = 2048 bytes. Thus the internal structure varies depending on which data type is used. Consequently, it is important to know whether the data type is with or without sound (or is only sound data) for conversion and when burning data onto the CD-ROM. Please refer to the STR data document for details regarding the respective data formats.

The following procedure is used to create video data without sound:

1. Prepare for conversion

Prepare the video data to be converted. The pre-conversion data may include sound. Subsequent settings can be used to convert only the image component. mc32 does not convert the image size or the frame rate, therefore, it is necessary to convert these attributes separately before inputting the data into mc32. This conversion can be performed using a general editing tool. As required, the images can also be filtered using a tool such as a low-pass filter in order to maintain high image quality after compression.

2. Start up mc32

Launch mc32. When mc32 starts up, the Conversion window will open automatically.

3. Set up the Conversion window

In the Conversion window, set the input filename, input format, and the output filename. Specify STR(MDEC) for the output format.

4. Set up the MDEC Parameters window

Clear the Sound check box in the Output Type so that the output data will be without sound. This check box is only for specifying the output type, regardless of whether the input data does or does not have sound. Even if the input data contains sound, the sound will not be converted if the Sound check box has been cleared. The MDEC Image Version should normally be specified as version 2.

If Easy Settings is selected, the frame rate, number of channels, CD-ROM playback speed, and leap sector can be set. For the frame rate, specify the frame rate during playback. Even if the frame rate of the input image is different from the specified frame rate, the conversion will still be performed. However, during playback, slow playback or speed playback will be indicated. Therefore, the frame rate of the input image should be converted separately to the playback frame rate.

Specify 1 ch for the number of channels. The playback speed of the CD-ROM will set the playback time value. Set or clear the leap sector depending upon whether the input images are material captured from an image medium or were created by means of CG rendering. For further information, refer to the "MDEC Parameters Window" subsection.

If Custom Settings is selected, set the maximum frame size and the frame integration method.

5. Set up the MDEC Parameter - Q Matrix window

The MDEC Parameter - Q Matrix window can be used to update a value in the quantization matrix. Normally, this is not necessary.

6. Check image quality check in the Video window

After the filename is specified using the File Input dialog box in the Conversion window, the Video window will be opened automatically. The Video window can also be opened by specifying the input file in the File/Open menu. Use the Encode button in the Video window to compress/expand the displayed

image and check the image quality. If there is a problem with the image quality, make adjustments in the MDEC Parameters window and recheck the image.

7. Convert

If the image quality is acceptable, perform the conversion in the Conversion window. Press the Cancel button to cancel the conversion.

Creating Video Data with Sound (Simultaneous Conversion of Sound)

There are two ways to create video data with sound. One method involves converting AVI with sound to video with sound, simultaneously converting the video and sound components. The other method converts the video and sound components separately, then mixes them together after conversion.

In order to simultaneously convert video and sound, the input data must include sound. The processing steps are almost identical to the procedure for creating video data without sound, except that the Sound check box must be checked in the Output Type setting of the MDEC Parameters window. Note that when the input video contains sound and this check box is set, sound conversion and video/sound synthesis are performed automatically after the video is converted.

Creating Video Data with Sound (Separate Conversion of Sound)

When the input video data does not contain sound, it is necessary to perform video conversion, sound conversion, and synthesis separately.

1. Create the Video Data

The procedure for creating video data is almost identical to that for creating video data without sound. The only difference is that the Sound check box must be checked in the Output Type setting of the MDEC Parameters window.

When the input video does not contain sound and this check box is set, the video is compressed after deducting the size of the sound component. This assumes that the video will ultimately be combined with the sound. Because the input video does not contain sound, care must be taken to avoid accidentally clearing the Sound check box. The Sound check box is used to specify sound / no sound for the final output video, therefore, if the Sound check box is cleared by mistake, conversion will be performed and the video will be synthesized with the sound, but the final video and sound will not be synchronized.

2. Create the Sound Data

The sound data creation procedure is exactly the same as the procedure for creating sound data that is described in the next section. The sound artist tool (RAW2XA) can be used for sound conversion instead of mc32.

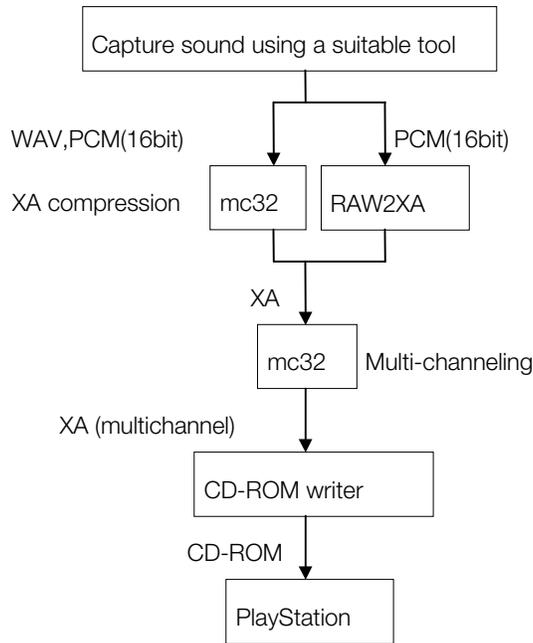
3. Synthesize the Video with the Sound

Synthesis can be performed after entering the required items (e.g., input video, input sound, output filename, conversion parameters, etc.) in the Video+Sound window.

Creating Sound Data

The following figure shows the processing steps from sound capture through creation of XA sound data.

Figure 1-13: Sound Data Creation Procedure



Sound data captured from a suitable tool is stored in WAV or PCM (16-bit) format and is converted to XA sound using mc32. The conversion can also be performed using the sound artist tool RAW2XA. The resulting XA sound data is then converted to multiple channels using mc32. Finally, a CD-ROM writer is used to burn the data onto a CD-ROM. The sound data will not play back correctly if it is burned directly onto the CD-ROM without first being converted to multiple channels.

The conversion of XA sound using mc32 is explained in the following subsection. For information on how to burn a CD-ROM using a CD-ROM writer, refer to the subsection entitled "Burning Streaming Data into a CD-ROM." Note that sound data to be converted (WAV, PCM) must have a sample length of 16 bits.

In the following section, window functions are only briefly discussed. For more detailed information, refer to the section entitled "mc32 Operation."

1. Start up mc32

Launch mc32. When mc32 starts up, the Conversion window will open automatically.

2. Set up the Conversion window

In the Conversion window, set the input filename, input format, and the output filename. Specify XA for the output format.

3. Set up the XA Parameters window

In the XA Parameters window, set stereo or monaural mode and the frequency of the output XA sound. If the frequency of the input XA sound differs from the specified output frequency, the frequency will be converted automatically.

4. Convert the sound

Perform the conversion in the Conversion window. Press the Cancel button to cancel the conversion.

5. Set up the Multi Channel window

Follow the procedure described in the "Multi Channel Window" subsection to create multiple channels of XA sound. Since XA sound is a subheader file format with 2336-bytes/sector, check the Sub Header check box for both input and output.

Creating Multichannel Data

Multichannel Streaming

Multichannel streaming is a streaming format used to interleave multiple streams of data on a CD-ROM. In ordinary streaming, one stream of data occupies contiguous regions on the CD-ROM. In multichannel streaming, however, multiple channels are interleaved in contiguous regions on the CD-ROM. Each channel is given a channel number, so it is possible to play only a specific channel during playback.

Figure 1-14: Ordinary Streaming

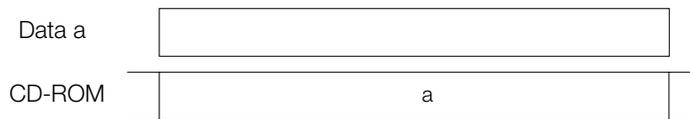
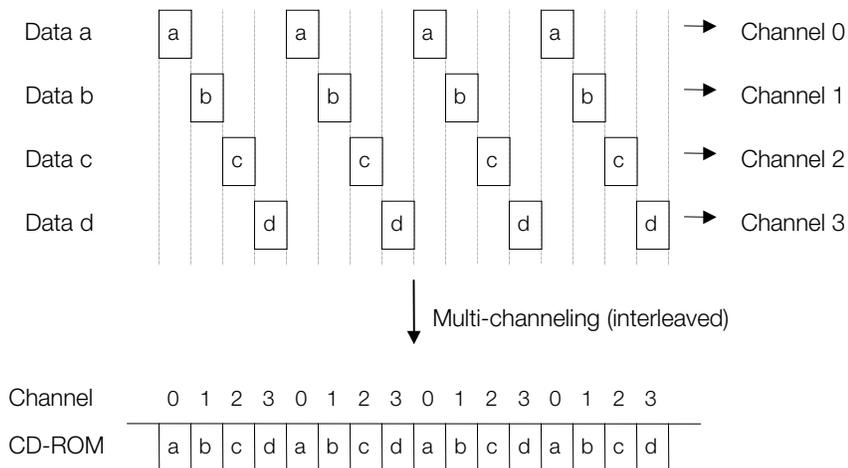


Figure 1-15: Multi-channel Streaming



Multichannel streaming has some important advantages: CD-ROM storage can be used efficiently, and playback data can be switched at high speed.

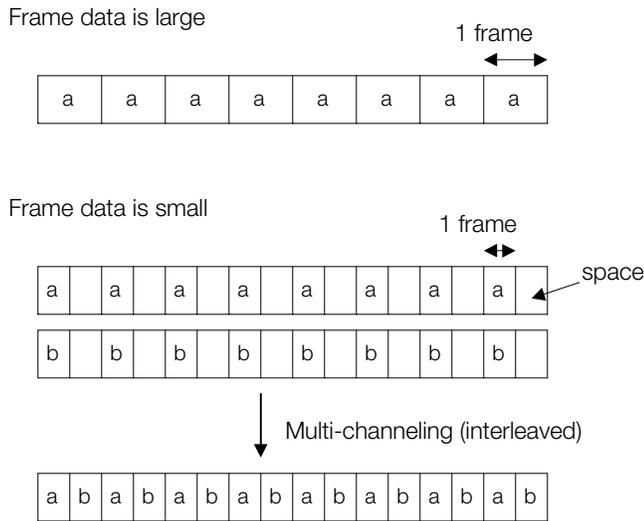
Efficient Use of CD-ROM Storage

In ordinary streaming (without multiple channels), the amount of CD-ROM storage needed depends only on the playback time of the streaming data. In other words, for a given playback time, all streaming data require exactly the same amount of storage on the CD-ROM. For example, for a 16x16-size video and a 640x240-size video, if playback time is the same for both, the amount of CD-ROM storage required will also be the same. This is because the PlayStation's streaming mechanism operates in sync with the reading of data from the CD-ROM.

However, if the storage areas on the CD-ROM are examined more closely, it is evident that not all of these areas are used efficiently. For example, for a 16x16-size video, large spaces are inserted between each frame's data in order to correctly maintain playback timing. Generally speaking, if frame data is sparse, the spaces are large. If there is a lot of frame data, the spaces are small. In the case of 16x16-size video, the frame data is so sparse that the spaces themselves become much larger than the actual data. In

multichannel streaming, this space contains other streaming data. Consequently, multichannel streaming can use CD-ROM storage more efficiently than ordinary streaming.

Figure 1-16: Efficient use of CD-ROM storage with multiple channels



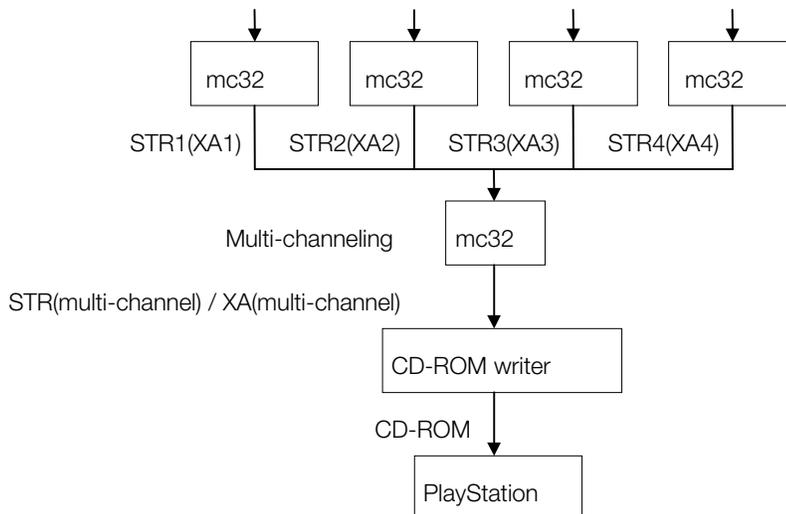
High-Speed Switching of Playback Data

In multichannel streaming, it is possible to switch to other streaming data by merely changing the specified channel to be played back. Because a CD-ROM head seek is not required, switching is very fast. Switching can also be performed during playback.

Creating Multichannel Data

The processing steps from the creation of data for each channel through the creation of multiple channels of data are shown below.

Figure 1-17: Procedure for creating multi-channel data



1. Start up mc32

Launch mc32. When mc32 starts up, the Conversion window will open automatically.

2. Create data for each channel using the Conversion window

Multiple channels of MDEC video and XA sound can be created in mc32. This is done by creating multiple channels in advance, then converting the data into the appropriate bands.

For example, in order to perform four channels of multichannel streaming during double-speed playback, the band allowed per channel is calculated as follows:

$$300(\text{KByte/sec}) / 4 = 75(\text{KBytes/sec})$$

(double-speed CD-ROM data transfer rate: 300 Kbytes/sec)

For MDEC video, the desired values for the total number of channels for conversion should be set in the MDEC Parameters window. For XA audio, the total number of channels is determined automatically according to the CD-ROM speed, frequency, and whether the mode is stereo or monaural, so it does not need to be specified. For more information on the conversion of video and sound, please refer to the sections entitled "Creation of Video" and "Creation of Sound," respectively.

3. Set up the Multi Channel window

The total number of channels specified in the Multi Channel window depends on the type of data to be processed. For MDEC video, use the total number of channels that were specified during channel data creation. For XA sound, use the total number of channels as described in the subsection entitled "Multi Channel Window."

4. Create multiple channels

Perform the conversion in the Multi Channel window. Press the Cancel button to cancel the conversion.

Scripting

mc32 has a function which allows conversions to be performed in batch mode. A script file describes the batch processing flow. Script files allows large numbers of conversions to be performed in a single operation. For more information on script editing and execution, please refer to the "Script Window" subsection.

The syntax of an mc32 script is very simple. Functions are executed sequentially, and there is no branching or looping. The functions that can be used within a script perform interleaving and conversion operations.

The following list contains all scripting functions available in mc32. The function names and their arguments are not case sensitive. Functions that end with AV are used to perform video/sound synthesis after video conversion. Functions that end in V are used when synthesis is not to be performed after video conversion. Functions that end in C are used when the frame size is specified directly. For example, when the input is AVI with sound, `Avi2strMdecAV()` converts AVI video to STR(MDEC), automatically converts the sound, then automatically synthesizes the video together with the sound.

If a line begins with a # symbol it is considered to be a comment.

Table 1-6: Script Functions

Function	Description
Avi2strMdecAv()	Creates STR(MDEC) from AVI (Easy Settings).
Avi2strMdecV()	Creates STR(MDEC) from AVI (Easy Settings).
Avi2strMdecC()	Creates STR(MDEC) from AVI (Custom Settings).
Avi2bsAv()	Creates BS from AVI (Easy Settings).
Avi2bsV()	Creates BS from AVI (Easy Settings).
Avi2bsC()	Creates BS from AVI (Custom Settings).
Yuv2strMdecAv()	Creates STR(MDEC) from YUV (Easy Settings).
Yuv2strMdecV()	Creates STR(MDEC) from YUV (Easy Settings).
Yuv2strMdecC()	Creates STR(MDEC) from YUV (Custom Settings).
Yuv2bsAv	Creates BS from YUV (Easy Settings).
Yuv2bsV	Creates BS from YUV (Easy Settings).
Yuv2bsC	Creates BS from YUV (Custom Settings).
Yuv2Avi	Creates AVI from YUV.
Rgb2strMdecAv()	Creates STR(MDEC) from RGB (Easy Settings).
Rgb2strMdecV()	Creates STR(MDEC) from RGB (Easy Settings).
Rgb2strMdecC()	Creates STR(MDEC) from RGB (Custom Settings).
Rgb2bsAv	Creates BS from RGB (Easy Settings).
Rgb2bsV	Creates BS from RGB (Easy Settings).
Rgb2bsC	Creates BS from RGB (Custom Settings).
Rgb2Avi	Creates AVI from RGB.
Tim2strMdecAv	Creates STR(MDEC) from TIM (Easy Settings).
Tim2strMdecV	Creates STR(MDEC) from TIM (Easy Settings).
Tim2strMdecC	Creates STR(MDEC) from TIM (Custom Settings).
Tim2bsAv	Creates BS from TIM (Easy Settings).
Tim2bsV	Creates BS from TIM (Easy Settings).
Tim2bsC	Creates BS from TIM (Custom Settings).
Wav2xa()	Creates XA from a WAV file.
Pcm2xa()	Creates XA from a PCM file.
MakeAv()	Synthesizes video with sound.
MakeV()	Converts the video format.
Pack1ch	Performs multichanneling (1 ch).
Pack2ch	Performs multichanneling (2 ch).
Pack4ch	Performs multichanneling (4 ch).
Pack8ch	Performs multichanneling (8 ch).
Pack16ch	Performs multichanneling (16 ch).
Pack32ch	Performs multichanneling (32 ch).
SetQ	Sets the quantization matrices.
SetDefaultQ	Sets the default values in the quantization matrices.

Avi2bsV(InputFilename, OutputFilename, CD-ROMspeed, FrameRate,NumberOfChannels, MDECversion, LeapSector)

```

[Function]                Creates BS from AVI (Easy Settings).
[Arguments]
    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    CD-ROMspeed:          Specifies the CD-ROM speed. (x1/x2)
    FrameRate:            Specifies the frame rate.
                          (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels:     Specifies the number of channels. (1/2/4)
    MDECversion:          Specifies the MDEC version. (2/3)
    LeapSector:           Specifies a leap sector. (TRUE/FALSE)

[Example]
Avi2bsV(
    c:\tmp\data.avi,      # Input file name
    c:\tmp\aviv0000.bs,  # Output file name
    x2,                   # CD-ROM speed
    15fps,                # Frame rate
    2,                    # Number of channels
    2,                    # MDEC version
    TRUE                  # LeapSector
);

```

Avi2bsC(InputFilename, OutputFilename, MaxFrameSize,FixedFrameSize, MDECversion)

```

[Function]                Creates BS from AVI (Custom Settings).
[Arguments]
    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    MaxFrameSize:         Specifies the maximum size of one frame.
                          ex. '10sectors', '20000bytes'
    FixedFrameSize:       Specifies whether the frame size is fixed or
                          variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:          Specifies the MDEC version. (2/3)

[Example]
Avi2bsC(
    c:\tmp\data.avi,      # Input file name
    c:\tmp\avic0000.bs,  # Output file name
    10sectors,           # Maximum size for one frame
    FixedFrameSize,      # Fixed frame size or not
    2                    # MDEC version
);

```



```
[Example]
Yuv2bsV(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\yuvv0000.bs, # Output file name
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    2,                   # Number of channels
    2,                   # MDEC version
    TRUE                 # LeapSector
);
```

Yuv2bsC (InputFilename, Width, Height, OutputFilename, MaxFrameSize, FixedFrameSize, MDECversion)

```
[Function]           Creates BS from YUV (Custom Settings).
[Arguments]
    InputFilename:    Specifies the input filename.
    Width:            Specifies the width of the input image.
    Height:           Specifies the height of the input image.
    OutputFilename:   Specifies the output filename.
    MaxFrameSize:     Specifies the maximum size of one frame.
                    ex. '10sectors', '20000bytes'
    FixedFrameSize:   Specifies whether the frame size is fixed or
                    variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:      Specifies the MDEC version. (2/3)
```

```
[Example]
Yuv2bsC(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\yuvc0000.bs, # Output file name
    10sectors,          # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                    # MDEC version
);
```

Yuv2Avi (InputFilename, Width, Height, OutputFilename, FrameRate)

```
[Function]           Creates AVI from YUV.
[Arguments]
    InputFilename:    Specifies the input filename.
    Width:            Specifies the width of the input image.
    Height:           Specifies the height of the input image.
    OutputFilename:   Specifies the output filename.
    FrameRate:        Specifies the frame rate.
```

```
[Example]
Yuv2avi(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\yuv.avi,     # Output file name
    15fps                # Frame rate
);
```

Rgb2strMdecAv(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[Function] Creates STR(MDEC) from RGB (Easy Settings).

[Arguments]

InputFilename: Specifies the input filename.
 Width: Specifies the width of the input image.
 Height: Specifies the height of the input image.
 OutputFilename: Specifies the output filename.
 CD-ROMspeed: Specifies the CD-ROM speed. (x1/x2)
 FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 MDECversion: Specifies the MDEC version. (2/3)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)
 SoundFrequency: Selects the XA sound frequency. (37.8KHz/18.9KHz)
 StereoMono: Specifies stereo or monaural. (Stereo/Mono)

[Example]

```
Rgb2strMdecAv(
  c:\tmp\data0000.rgb, # Input file name
  320,                 # Width
  240,                 # Height
  c:\tmp\out7.str,    # Output file name
  x2,                  # CD-ROM speed
  15fps,               # Frame rate
  1,                   # Number of channels
  2,                   # MDEC version
  TRUE,                # LeapSector
  37.8KHz,             # Frequency of Sound
  Mono                 # Stereo or Mono
);
```

Rgb2strMdecV(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector)

[Function] Creates STR(MDEC) from RGB (Easy Settings).

[Arguments]

InputFilename: Specifies the input filename.
 Width: Specifies the width of the input image.
 Height: Specifies the height of the input image.
 OutputFilename: Specifies the output filename.
 CD-ROMspeed: Specifies the CD-ROM speed. (x1/x2)
 FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 MDECversion: Specifies the MDEC version. (2/3)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)

[Example]

```

Rgb2strMdecV(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out8.str,    # Output file name
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    2,                   # Number of channels
    2,                   # MDEC version
    TRUE                 # LeapSector
);

```

Rgb2strMdecC(InputFilename, Width, Height, OutputFilename, MaxFrameSize, FixedFrameSize, MDECversion)

[Function] Creates STR(MDEC) from RGB (Custom Settings).

[Arguments]

InputFilename:	Specifies the input filename.
Width:	Specifies the width of the input image.
Height:	Specifies the height of the input image.
OutputFilename:	Specifies the output filename.
MaxFrameSize:	Specifies the maximum size of one frame. ex. '10sectors', '20000bytes'
FixedFrameSize:	Specifies whether the frame size is fixed or variable. ('FixedFrameSize/VariableFrameSize')
MDECversion:	Specifies the MDEC version. (2/3)

[Example]

```

Rgb2strMdecC(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out9.str,    # Output file name
    10sectors,          # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                    # MDEC version
);

```

Rgb2bsAv (InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[Function] Creates BS from RGB (Easy Settings).

[Arguments]

InputFilename:	Specifies the input filename.
Width:	Specifies the width of the input image.
Height:	Specifies the height of the input image.
OutputFilename:	Specifies the output filename.
CD-ROMspeed:	Specifies the CD-ROM speed. (x1/x2)
FrameRate:	Specifies the frame rate. (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
NumberOfChannels:	Specifies the number of channels. (1/2/4)
MDECversion:	Specifies the MDEC version. (2/3)
LeapSector:	Specifies a leap sector. (TRUE/FALSE)
SoundFrequency:	Selects the XA sound frequency. (37.8KHz/18.9KHz)
StereoMono:	Specifies stereo or monaural. (Stereo/Mono)

```
[Example]
Rgb2bsAv(
    c:\tmp\data0000.rgb,# Input file name
    320,                # Width
    240,                # Height
    c:\tmp\rgba0000.bs,# Output file name
    x2,                 # CD-ROM speed
    15fps,              # Frame rate
    1,                  # Number of channels
    2,                  # MDEC version
    TRUE,               # LeapSector
    37.8KHz,           # Frequency of Sound
    Mono                # Stereo or Mono
);
```

Rgb2bsV(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector)

```
[Function]                Creates BS from RGB (Easy Settings).
[Arguments]
    InputFilename:         Specifies the input filename.
    Width:                 Specifies the width of the input image.
    Height:                Specifies the height of the input image.
    OutputFilename:        Specifies the output filename.
    CD-ROMspeed:           Specifies the CD-ROM speed. (x1/x2)
    FrameRate:             Specifies the frame rate.
                          (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels:      Specifies the number of channels. (1/2/4)
    MDECversion:           Specifies the MDEC version. (2/3)
    LeapSector:            Specifies a leap sector. (TRUE/FALSE)
```

```
[Example]
Rgb2bsV(
    c:\tmp\data0000.rgb,# Input file name
    320,                # Width
    240,                # Height
    c:\tmp\rgbv0000.bs,# Output file name
    x2,                 # CD-ROM speed
    15fps,              # Frame rate
    2,                  # Number of channels
    2,                  # MDEC version
    TRUE                # LeapSector
);
```

Rgb2bsC (InputFilename, Width, Height, OutputFilename, MaxFrameSize, FixedFrameSize, MDECversion)

```

[Function]                Creates BS from RGB (Custom Settings).
[Arguments]
    InputFilename:        Specifies the input filename.
    Width:                Specifies the width of the input image.
    Height:               Specifies the height of the input image.
    OutputFilename:       Specifies the output filename.
    MaxFrameSize:         Specifies the maximum size of one frame.
                        ex. '10sectors', '20000bytes'
    FixedFrameSize:       Specifies whether the frame size is fixed or
                        variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:          Specifies the MDEC version. (2/3)

[Example]
Rgb2bsC(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\rgbc0000.bs, # Output file name
    10sectors,           # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                    # MDEC version
);

```

Rgb2Avi (InputFilename, Width, Height, OutputFilename, FrameRate)

```

[Function]                Creates AVI from RGB.
[Arguments]
    InputFilename:        Specifies the input filename.
    Width:                Specifies the width of the input image.
    Height:               Specifies the height of the input image.
    OutputFilename:       Specifies the output filename.
    FrameRate:            Specifies the frame rate.

[Example]
Rgb2avi(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\rgb.avi,      # Output file name
    15fps                 # Frame rate
);

```

Tim2strMdecAv(InputFilename, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[Function] Creates STR(MDEC) from TIM (Easy Settings).

[Arguments]

InputFilename: Specifies the input filename.
 OutputFilename: Specifies the output filename.
 CD-ROMspeed: Specifies the CD-ROM speed. (x1/x2)
 FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 MDECversion: Specifies the MDEC version. (2/3)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)
 SoundFrequency: Selects the XA sound frequency. (37.8KHz/18.9KHz)
 StereoMono: Specifies stereo or monaural. (Stereo/Mono)

[Example]

```
Tim2strMdecAv(
  c:\tmp\data0000.tim, # Input file name
  c:\tmp\out1.str,    # Output file name
  x2,                 # CD-ROM speed
  15fps,              # Frame rate
  1,                  # Number of channels
  2,                  # MDEC version
  FALSE,              # LeapSector
  37.8KHz,            # Frequency of Sound
  Stereo              # Stereo or Mono
);
```

Tim2strMdecV(InputFilename, OutputFilename, CD-ROMspeed, FrameRate,NumberOfChannels, MDECversion, LeapSector)

[Function] Creates STR(MDEC) from TIM (Easy Settings).

[Arguments]

InputFilename: Specifies the input filename.
 OutputFilename: Specifies the output filename.
 CD-ROMspeed: Specifies the CD-ROM speed. (x1/x2)
 FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 MDECversion: Specifies the MDEC version. (2/3)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)

[Example]

```
Tim2strMdecV(
  c:\tmp\data0000.tim, # Input file name
  c:\tmp\out2.str,    # Output file name
  x2,                 # CD-ROM speed
  15fps,              # Frame rate
  2,                  # Number of channels
  2,                  # MDEC version
  TRUE                # LeapSector
);
```


Tim2bsV(InputFilename, OutputFilename, CD-ROMspeed, FrameRate,NumberOfChannels, MDECversion, LeapSector)

```

[Function]                Creates BS from TIM (Easy Settings).
[Arguments]
    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    CD-ROMspeed:          Specifies the CD-ROM speed. (x1/x2)
    FrameRate:            Specifies the frame rate.
                          (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels:     Specifies the number of channels. (1/2/4)
    MDECversion:          Specifies the MDEC version. (2/3)
    LeapSector:           Specifies a leap sector. (TRUE/FALSE)

[Example]
Tim2bsV(
    c:\tmp\data0000.tim,# Input file name
    c:\tmp\timv0000.bs, # Output file name
    x2,                 # CD-ROM speed
    15fps,              # Frame rate
    2,                  # Number of channels
    2,                  # MDEC version
    TRUE                # LeapSector
);

```

Tim2bsC(InputFilename, OutputFilename, MaxFrameSize,FixedFrameSize, MDECversion)

```

[Function]                Creates BS from TIM (Custom Settings).
[Arguments]
    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    MaxFrameSize:         Specifies the maximum size of one frame.
                          ex. '10sectors', '20000bytes'
    FixedFrameSize:       Specifies whether the frame size is fixed or
                          variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:          Specifies the MDEC version. (2/3)

[Example]
Tim2bsC(
    c:\tmp\data0000.tim,# Input file name
    c:\tmp\avic0000.bs, # Output file name
    10sectors,          # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                    # MDEC version
);

```

Wav2xa(InputFilename, OutputFilename, SoundFrequency, StereoMono)

```

[Function]                Creates XA from a WAV file.
[Arguments]
    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    SoundFrequency:       Selects the XA sound frequency. (37.8KHz/18.9KHz)
    StereoMono:           Specifies stereo or monaural. (Stereo/Mono)
[Example]
Wav2xa (
    c:\tmp\data.wav,      # Input file name
    c:\tmp\data.xa,      # Output file name
    37.8KHz,             # Frequency of xa Sound
    Stereo                # Stereo or Mono for xa Sound
);

```

Pcm2xa(InputFilename, PCMBitsPerSample, PCMFrequency, PCMstereoMono, OutputFilename, XAFrequency, XAstereoMono)

```

[Function]                Creates XA from a PCM file.
[Arguments]
    InputFilename:        Specifies the input filename.
    PCMBitsPerSample:     Specifies the sample length of the PCM data.
                          (16bit)
    PCMFrequency:         Specifies the frequency of the PCM data.
                          (unit: KHz)
    PCMstereoMono:        Specifies stereo or monaural PCM data.
                          (Stereo/Mono)
    OutputFilename:       Specifies the output filename.
    XAFrequency:          Selects the XA sound frequency. (37.8KHz/18.9KHz)
    XAstereoMono:         Specifies stereo or monaural. (Stereo/Mono)
[Example]
Pcm2xa (
    c:\tmp\data.pcm,     # Input file name
    16bit,               # Bits Per Sample
    44.1KHz,             # Frequency of PCM Sound (KHz)
    Stereo,              # Stereo or Mono for PCM Sound
    c:\tmp\data.xa,     # Output file name
    37.8KHz,             # Frequency of xa Sound
    Stereo                # Stereo or Mono for xa Sound
);

```

MakeAv(VideoFilename, VideoFormatName, SoundFileName, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, LeapSector, OldLeapSector, SoundFrequency, StereoMono)

```

[Function]                Synthesizes video with sound.
[Arguments]
    VideoFilename:        Specifies the input filename.
    VideoFormatName:      Specifies the image format name. (strMDEC/bs)
    SoundFileName:        Specifies the sound filename.
    OutputFilename:       Specifies the output filename.
    CD-ROMspeed:          Specifies the CD-ROM speed. (x1/x2)

```

FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)
 OldLeapSector: Specifies whether or not the leap sector version
 is old.
 Specify TRUE if the image was created with a
 version of MovConv before version 2.0. Otherwise,
 specify FALSE. (TRUE/FALSE)
 SoundFrequency: Selects the XA sound frequency. (37.8KHz/18.9KHz)
 StereoMono: Specifies stereo or monaural. (Stereo/Mono)

[Example]

```

MakeAv(
  c:\tmp\data.str,      # Video file name
  strMDEC,             # Video format name
  c:\tmp\data.xa,      # Sound file name
  c:\tmp\out10.str,    # Output file name (Sound & video)
  x2,                  # CD-ROM speed
  15fps,               # Frame rate
  1,                   # Number of channels
  FALSE,               # LeapSector
  FALSE,               # ver1.98 compatible leap sector
  37.8KHz,             # Frequency of Sound
  Stereo                # Stereo or Mono
);
  
```

MakeV(VideoFilename, VideoFormatName, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, LeapSector, OldLeapSector)

[Function] Converts the video format.

[Arguments]

VideoFilename: Specifies the input filename.
 VideoFormatName: Specifies the video format name. (strMDEC/bs)
 OutputFilename: Specifies the output filename.
 CD-ROMspeed: Specifies the CD-ROM speed. (x1/x2)
 FrameRate: Specifies the frame rate.
 (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
 NumberOfChannels: Specifies the number of channels. (1/2/4)
 LeapSector: Specifies a leap sector. (TRUE/FALSE)
 OldLeapSector: Specifies whether or not the leap sector version
 is old.
 Specify TRUE if the image was created with a
 version of MovConv before version 2.0. Otherwise,
 specify FALSE. (TRUE/FALSE)

[Example]

```

MakeV(
  c:\tmp\data.str,      # Video file name
  strMDEC,             # Video format name
  c:\tmp\out11.str,    # Output file name (video)
  x2,                  # CD-ROM speed
  15fps,               # Frame rate
  1,                   # Number of channels
  TRUE,                # LeapSector
  FALSE                # ver1.98 compatible leap sector
);
  
```


ChTermType0-1: Specifies the terminator type for each channel.
(0:No data/1:Video sector/2:Null sector)

ChTermLength0-1: Specifies the data length for each channel's
terminator data.

This value is significant only when ChTermType is
1 or 2. (unit: sector)

[Example]

```
Pack2ch(
  # output file      [subheader]
  d:\tmp\out.str,   FALSE,

  #
  # channel data files (xa or str)
  #
  # [filename]      [subheader] [term type] [length]
  #                TRUE/FALSE  (0 - 2)
  #
  d:\tmp\data0.str, FALSE,      0,          0,
  d:\tmp\data1.str, FALSE,      0,          0
);
```

**Pack4ch(OutputFilename, OutputSubheader,
ChFileName0, ChSubheader0, ChTermType0, ChTermLength0,
ChFileName1, ChSubheader1, ChTermType1, ChTermLength1,
ChFileName2, ChSubheader2, ChTermType2, ChTermLength2,
ChFileName3, ChSubheader3, ChTermType3, ChTermLength3
)**

[Function] Creates multiple channels of STR data or XA
data. (4ch)

For details regarding the setting method, refer
to the "Multi Channel Window" subsection.

[Arguments]

OutputFilename: Specifies the output filename.

OutputSubheader: Sets the subheader attribute of the output
data. (TRUE/FALSE)

ChFileName0-3: Specifies the filename for each channel data.
'None' is specified for an empty channel.

ChSubheader0-3: Sets the subheader attribute for each channel
data. (TRUE/FALSE)

ChTermType0-3: Specifies the terminator type for each channel.
(0:No data/1:Video sector/2:Null sector)

ChTermLength0-3: Specifies the data length for each channel's
terminator data.

This value is significant only when ChTermType is
1 or 2. (unit: sector)

```
[Example]
Pack4ch(
    # output file          [subheader]
    d:\tmp\out.xa,        FALSE,

    #
    # channel data files (xa or str)
    #
    # [filename]          [subheader] [term type] [length]
    #                    TRUE/FALSE  (0 - 2)
    #
    d:\tmp\data0.xa,     FALSE,          0,          0,
    d:\tmp\data1.xa,     FALSE,          0,          0,
    d:\tmp\data2.xa,     FALSE,          0,          0,
    d:\tmp\data3.xa,     FALSE,          0,          0
);
```

```
Pack8ch(OutputFilename, OutputSubheader,
ChFileName0, ChSubheader0, ChTermType0, ChTermLength0,
.....
ChFileName7, ChSubheader7, ChTermType7, ChTermLength7
)
```

[Function] Creates multiple channels of STR data or XA data. (8ch)

For details regarding the setting method, refer to the "Multi Channel Window" subsection.

[Arguments]

OutputFilename: Specifies the output filename.

OutputSubheader: Sets the subheader attribute of the output data. (TRUE/FALSE)

ChFileName0-7: Specifies the filename for each channel data. 'None' is specified for an empty channel.

ChSubheader0-7: Sets the subheader attribute for each channel data. (TRUE/FALSE)

ChTermType0-7: Specifies the terminator type for each channel. (0:No data/1:Video sector/2:Null sector)

ChTermLength0-7: Specifies the data length for each channel's terminator data.

This value is significant only when ChTermType is 1 or 2. (unit: sector)

[Example]

```
Pack8ch(
    # output file          [subheader]
    d:\tmp\out.xa,        FALSE,

    #
    # channel data files (xa or str)
    #
    # [filename]          [subheader] [term type] [length]
    #                    TRUE/FALSE  (0 - 2)
    #
    d:\tmp\data0.xa,     FALSE,          0,          0,
    .....
    d:\tmp\data7.xa,     FALSE,          0,          0
);
```


ChSubheader0-31: Sets the subheader attribute for each channel data. (TRUE/FALSE)

ChTermType0-31: Specifies the terminator type for each channel. (0:No data/1:Video sector/2:Null sector)

ChTermLength0-31: Specifies the data length for each channel's terminator data.

This value is significant only when ChTermType is 1 or 2. (unit: sector)

[Example]

```
Pack32ch(
    # output file          [subheader]
    d:\tmp\out.xa,        FALSE,

    #
    # channel data files (xa or str)
    #
    # [filename]          [subheader] [term type] [length]
    #                    TRUE/FALSE  (0 - 2)
    #
    d:\tmp\data0.xa,     FALSE,        0,            0,
    .....
    d:\tmp\data31.xa,   FALSE,        0,            0
);
```

SetQ(

Y_C,
coef00, coef01, coef02, coef03, coef04, coef05, coef06, coef07,
.....,
.....,
coef70, coef71, coef72, coef73, coef74, coef75, coef76, coef77,

)

[Function] Sets the quantization matrices.

[Arguments]

Y_C: Y is specified for the luminance block.
C is specified for the color-difference block.

coef00-coef77: Sets the quantization matrix coefficients.

[Example]

```
SetQ(
    Y,
    2,16,19,22,26,27,29,34,
    16,16,22,24,27,29,34,37,
    19,22,26,27,29,34,34,38,
    22,22,26,27,29,34,37,40,
    22,26,27,29,32,35,40,48,
    26,27,29,32,35,40,48,58,
    26,27,29,34,38,46,56,69,
    27,29,35,38,46,56,69,83
);
```

SetDefaultQ(Y_C)

[Function] Sets the default values in the quantization matrices.

[Arguments]

Y_C: Y is specified for the luminance block.
C is specified for the color-difference block.

[Example]

```
SetDefaultYQ(C);
```

Miscellaneous**QuickTime Data Conversion**

mc32 does not allow direct conversion from QuickTime movies. In order to convert from QuickTime, it is necessary to use a separate QuickTime --> AVI converter before using mc32.

```
QuickTime --> AVI --> STR
```

Burning Streaming Data onto a CD-ROM

The CD-ROM Generator is used to burn video data and sound data onto a CD-ROM. For this purpose, the CD-ROM Generator's parameters are set as follows:

When burning video data with sound:

File Type : Subheader File
Form 1 Sectors: on
Form 2 Sectors: on
Interleaved Sectors: on

When burning video data (data without a subheader) without sound:

File Type: Standard File

When burning XA sound data:

File Type: Subheader File
Form 1 Sectors: off
Form 2 Sectors: on
Interleaved Sectors: off

In order to position streaming data at a specific time on the CD-ROM, the location of the data must be specified in the CD-ROM Generator. In the CD-ROM Generator, location is specified as an absolute sector, and this can be determined from the absolute time using the following relationship:

$$\text{sector} = \text{time} * 75 - 150 \quad (\text{ex. } 60 \text{ seconds } \dots 4350 \text{ sectors})$$

For example, in order to place data at the 60-second position on a CD-ROM, specify sector 4350 as the location. For more detailed information on the use of the CD-ROM Generator, please refer to the CD-ROM Generator manual.

Capturing and Compressing High-Image-Quality Video

The quality of MDEC video played back on a PlayStation varies considerably depending upon the quality of the original video image, the efficiency of the capture equipment, the parameter settings during MDEC compression, and the compression preprocessing. Please take the following points into consideration when creating high-quality MDEC video.

Video Image Quality

The quality of the captured original video image greatly affects the quality of the MDEC video. Whenever possible, use high-quality media for the original video image. The image quality as a function of media is roughly shown below.

High Low
D1 > BETACAM > Hi8, S-VHS > VHS

If video does not need to pass through video equipment (e.g., video of CG-rendered data), images should be converted directly to RGB data or YUV data, then compressed using mc32. This prevents the deterioration of image quality that can occur from passing through video equipment.

Capture Equipment

Examples of capture equipment are videodisk recorders and PC video capture boards. Although PC video capture boards are inexpensive, the image quality they produce is generally poor. By contrast, the videodisk recorder enables high-quality video capture, but it is relatively expensive. Videodisk recorders use two capture methods: capturing and compressing in real time, and capturing without compression. For the purpose of optimizing image quality, capturing without compression is superior. However, if compression is not used, enormous amounts of memory are required and data transfer time is long.

Compression Parameters

MDEC compression is basically the same as JPEG compression. MDEC compression allows either the compression ratio or the image quality to be adjusted during compression. These two parameters are inversely related so when one is adjusted, the other will change automatically.

When streaming from the CD-ROM, the data transfer rate is determined from the speed of the CD-ROM drive. The data storage capacity allowed per frame is determined automatically from the frame rate.

mc32 can adjust the image quality during compression so that the compressed data uses the maximum amount of storage allowed by the frame rate. In mc32, the frame rate can be adjusted freely. When the frame rate is reduced, the amount of data per frame increases and image quality improves. Conversely, when the frame rate is increased, image quality deteriorates. Furthermore, image quality can vary considerably depending on the contents of the image, even when the frame rate remains constant.

Generally speaking, the more complex the image, the worse will be the image quality after compression. A complex image is one that contains many high-frequency components. This phenomenon is common to DCT compression methods and is not limited to MDEC compression.

Preprocessing

The greater the number of high-frequency components in an image, the worse will be the image quality for a given compression ratio. The quality of an image can be improved using filtering. Various types of filters exist, however, the low-pass filter, which removes high-frequency components, is extremely effective in improving image quality. When low-pass filtering is applied to an image after MDEC compression, noise is reduced below the level if no filter had been used.

Unfortunately, when low-pass filtering is applied to an image, its appearance will become less sharp. It therefore becomes necessary to select filter coefficients such that there is a proper balance between the degree of sharpness and the degree of noise.

Converting from AVI Data

A number of different frame compression formats can be used with AVI. mc32 fully supports compression formats that are host PC-compatible. However, compressed images are of lower quality and are consequently less desirable, therefore, uncompressed AVI should be used whenever possible.

MovConv: PlayStation Movie Converter

Version 2.2

Documentation date: 8/6/97

Description

MovConv is a movie converter for the PlayStation. MovConv provides the following features:

- Image conversion
- Sound conversion
- Linking of images and sound (interleaving)
- Execution of conversion and interleave scripts

MovConv handles the following image formats:

Input

Movie format: AVI (uncompressed)
 Static image sequences: RGB sequence (RGB files sequenced by filename)
 YUV sequence (YUV files sequenced by filename)
 TIM sequence (TIM files sequenced by filename)

Output

Movie formats: STR(MDEC)
 AVI (uncompressed)
 Static image sequences: BS sequence (BS file sequenced by filename)
 RGB sequence (RGB file sequenced by filename)

The STR (MDEC) and BS sequence output formats are data formats compressed for MDEC.

MovConv handles the following audio formats:

Input

WAV (Windows)
 PCM

Output

XA (CD-ROM XA)

When converting from WAV or PCM to XA, the sampling frequency and ADPCM compression are also converted.

Interleaving of images and sound is described below.

Input

STR(MDEC) or BS + XA

Output

STR(MDEC)

When converting data in which images and sounds are already linked, e.g. AVI with sound, interleaving will be performed automatically after the image and sound data are converted. However, if image and sound data are to be created separately, interleaving must be performed manually. Synchronization between images and sounds cannot be performed during interleaving, so the data must be synchronized before conversion.

The scripting feature permits batch processing of multiple conversion and interleave operations. Scripts must be written in the format defined by MovConv.

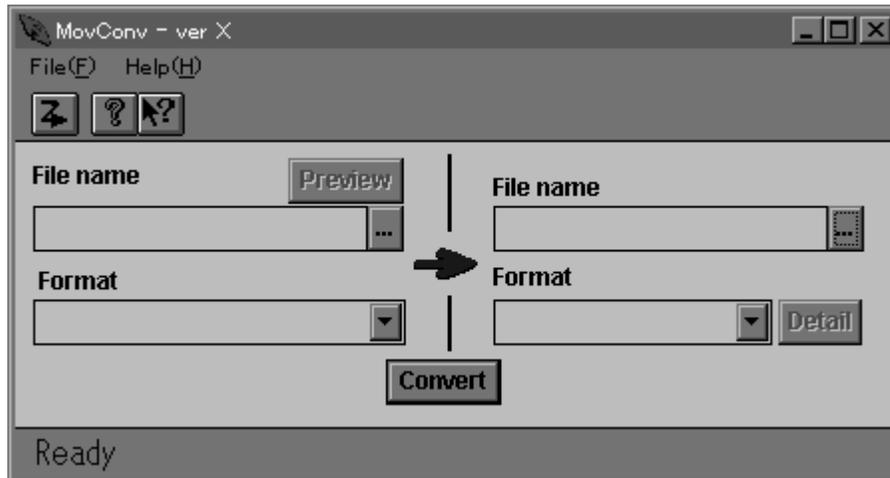
How to use

Please refer to the Movie Converter Tutorial for examples and specific information on using MovConv. The following section will focus on the various dialog boxes in MovConv.

The MovConv main window

The MovConv main window is the first window displayed after MovConv starts.

Figure 1-18: MovConv Main Window

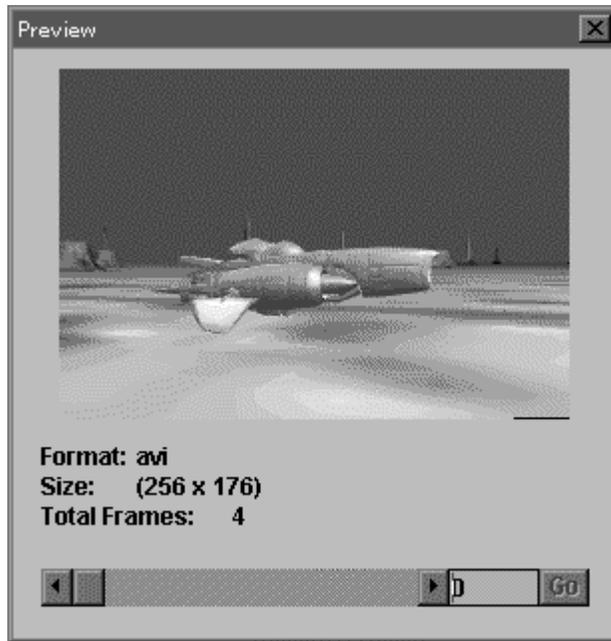


- Specifying an input filename**
 Clicking the left [...] button will open a File Open dialog box, where an input filename can be selected. A filename can also be entered directly in the left Filename text box.
- Specifying an input format**
 Once an input filename is entered, a format will be automatically displayed based on the filename extension. An input format can also be selected manually by using the left Format combo box.
- Specifying an output filename**
 Once an input filename and an input format are specified, an output filename will be displayed. To change the output filename, click on the right [...] button to open the File Save dialog box. A filename can also be entered directly into the right Filename text box.
- Specifying an output format**
 An output format is selected by using the right Format combo box.
- Output parameter settings**
 Once the output format has been selected, the [Detail] button can be clicked to display the Parameter Settings dialog box which corresponds to the output format.
- Previewing the input file**
 After correctly entering the input filename and the input format, click the [Preview] button or select Preview Input from the File menu. This will cause the Preview dialog box to be displayed.
- Performing the conversion**
 Once all the settings have been entered, click on the [Convert] button or select Convert from the File menu to perform the conversion.

The Preview dialog box

The Preview dialog box is opened by clicking the [Preview] button after the input file has been specified in the MovConv main window.

Figure 1-19: Preview Dialog Box

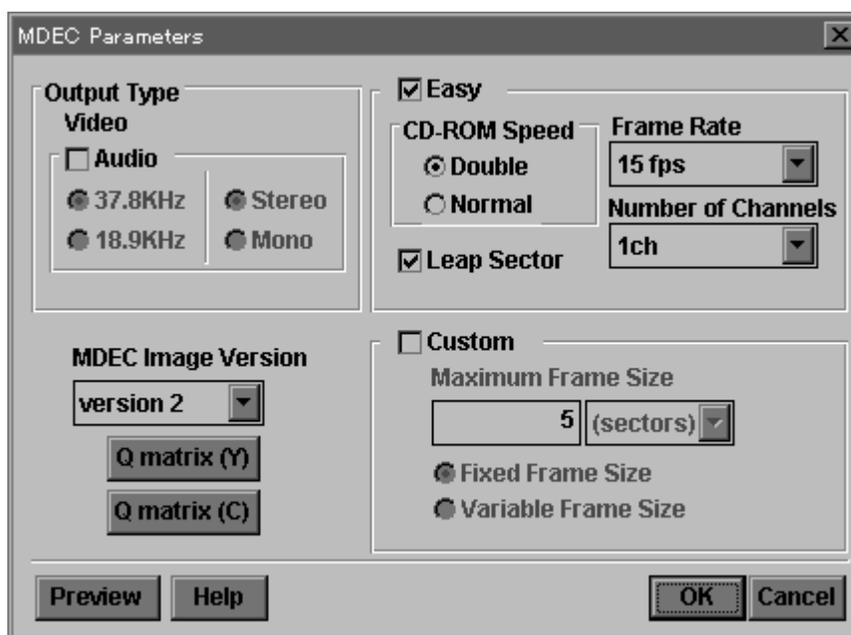


- Previewing the input file**
 The first frame of the input data is displayed in the dialog box. Use the scroll bar to jump to the target frame. Once the input data has been verified, click the Close button to exit Preview.

The MDEC Parameters dialog box

If the output format in the MovConv main window is set to "STR(MDEC)" or "BS", the MDEC Parameters dialog box can be displayed by clicking the [Detail] button.

Figure 1-20: MDEC Parameters Dialog Box



- **Selecting an output type**

Click the Audio check box if the output data includes sound (leave the box unchecked if there is no sound). This setting only applies to the output data and does not depend on whether the input data contains sound. The following table shows differences in the conversion operation as a function of the input data and output format settings.

Table 1-7: Conversion Differences as a Function of Audio

Input data	Output settings	Operation
Audio	Audio	Convert images, audio, and interleave image and audio data.
Audio	No Audio	Convert images only
No Audio	Audio	Convert images only. However, image conversion is performed so that space is left for audio. This setting can be used to convert images when image data and audio data are created separately and interleaved later.
No Audio	No Audio	Convert images only

- **Frequency, stereo/mono settings**

If audio is selected under Output Type, use the radio buttons in the Audio group to select output audio sampling frequencies and stereo/mono settings.

- **Selecting Easy/Custom Settings**

Parameter settings can be entered using Easy or Custom settings. When Easy is selected, settings are determined automatically as a function of post-compression frame size, CD-ROM playback speed, number of channels, and so on. Easy Settings should be used if special compressed frame sizes are not needed. With Custom Settings, the (maximum) size of compressed image frames can be specified by the user. Custom Settings permit images and movie frames to be compressed down to a certain size. If audio is selected under Output Type, only the Easy Settings method can be used. Check the Easy check box to select Easy Settings. Check the Custom check box to select Custom Settings.

- **Selecting CD-ROM playback speed**

If Easy Settings is selected, use the radio buttons in the CD-ROM Speed group to select double-speed or normal speed.

- **Selecting frame rates**

If Easy Settings is selected, use the Frame Rate combo box to select an appropriate frame rate.

- **Select number of channels**

If Easy Settings is selected, use the Number of Channels combo box to select an appropriate number of channels.

- **Leap sector setting**

If Easy Settings is selected, a leap sector can also be specified. Leap sectors are used to correct deviations that take place due to the difference between the 59.94fps frame rate used in video devices and the 60fps frame rate expected by the PlayStation for streaming from a CD-ROM. If leap sectors are enabled, an extra sector is inserted at regular intervals. The use of the leap sector setting is shown in the table below.

Table 1-8: Leap Sector Setting

Capture method	Leap Sector
Captured from video tape (59.94fps footage)	on
Footage with a frame rate of 60fps created using CG rendering, etc. (60fps footage)	off

- **Setting the maximum frame size**

If Custom Settings is selected, a maximum frame size for single compressed frames can be specified under Maximum Frame Size in the MDEC Parameters dialog box. Units can be specified in either sectors or bytes. The relationship between the two is as follows.

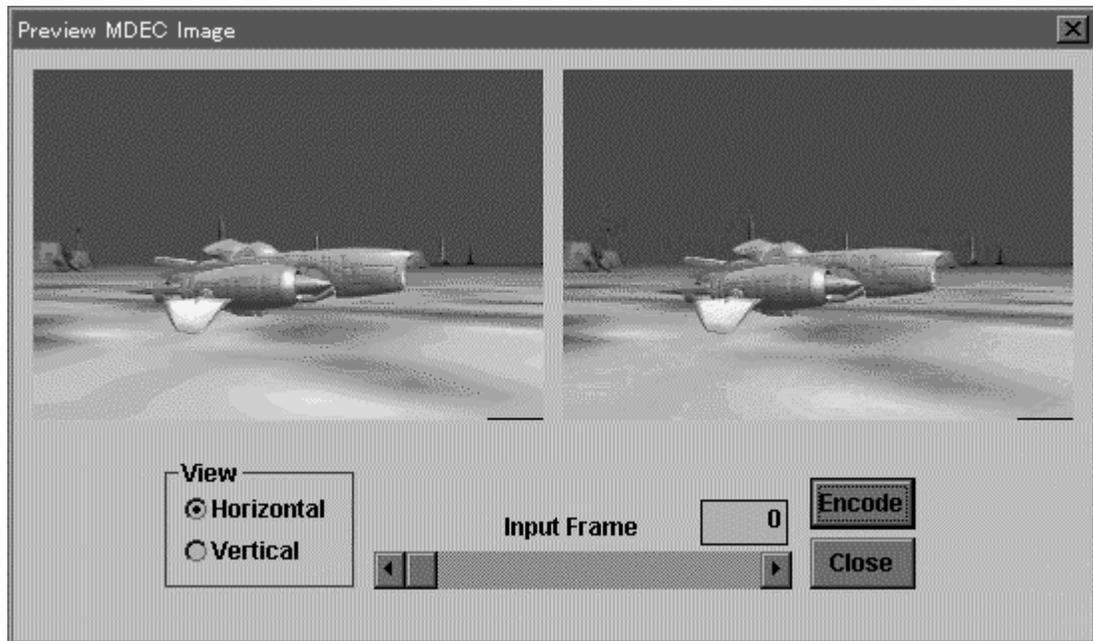
$$1 \text{ sector} = 2016 \text{ bytes}$$

- Settings for how frames are consolidated**
 If Custom Settings is selected, the method used in consolidating frames can be specified. The setting for maximum frame size specifies an upper bound on the compressed data size, so the actual result from compression may be less. If Fixed Frame Size is specified, dummy data will be inserted after the compressed data so that the resultant frame size will be at the maximum value. Dummy data will not be inserted if Variable Frame Size is specified.
- Selecting the compression format version**
 The version of the compression format to be used is selected using the MDEC Image Version combo box. Version 3 offers a slight improvement in image quality over version 2 but decoding is slower so version 2 is usually used.
- Quantization matrix settings**
 Clicking the [Q matrix(Y)] button and the [Q matrix(C)] button will display the Q matrix dialog box. The quantization matrix can be set using this dialog box. Apart from special cases, these settings do not need to be changed.
- Checking the compressed image quality**
 Clicking the [Preview] button in the MDEC Parameters dialog box will display the Preview MDEC Image dialog box. This dialog box can be used to check the image quality resulting from compression.

The Preview MDEC Image dialog box

Clicking the [Preview] button in the MDEC Parameters dialog box will display the Preview MDEC Image dialog box.

Figure 1-21: Preview MDEC Image Dialog Box

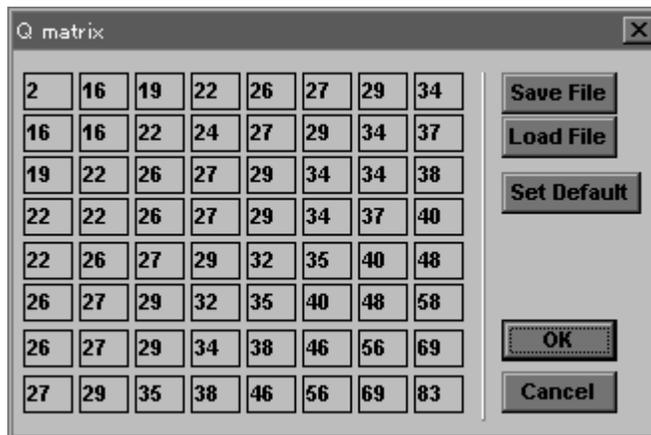


- Selecting a frame to be compressed**
 The scroll bar is used to select a frame to be compressed. A frame number can be entered directly into the Input Frame text box as well.
- Compression**
 Clicking the [Encode] button will compress the selected frame. A compressed image will then be displayed. The parameters used for the compression operation are those which were entered in the MDEC Parameters dialog box.
- Adjusting image position**
 If an image is positioned outside the viewing area, the Horizontal/Vertical radio buttons in the View group can be used to adjust the position.

The Q matrix dialog box

Clicking the [Q matrix(Y)] button or the [Q matrix(C)] button in the MDEC Parameters dialog box will display the Q matrix dialog box. Changing settings here will change the quantization matrix used during encoding. If [Q matrix(Y)] was clicked, the quantization matrix used in the luminance block will be changed. If [Q matrix(C)] was clicked, the quantization matrix used in the chrominance block will be changed. These settings can be changed to optimize the image quality for a particular image, but the default values should generally be adequate. If settings are changed, the same quantization matrix must be used for decoding. libpress provides a function for doing this. This function should be used to make the same changes to the quantization matrix during decoding. The default settings for the quantization matrix are the same as those used in the MPEG standard. For detailed information on the quantization matrix, please refer to references on MPEG, JPEG, etc.

Figure 1-22: Q Matrix Dialog Box



- **Changing the quantization matrix**
The values displayed in the rows and columns can be edited directly.
- **Default quantization matrix settings**
Clicking the [Set Default] button will return the quantization matrix settings to the default values.
- **Saving the quantization matrix**
Clicking the [Save File] button will dump the contents of the quantization matrix to a file.
- **Loading the quantization matrix**
Clicking the [Load File] button will load the contents of the quantization matrix from a file.

The XA Audio Parameters dialog box

Clicking the [Detail] button will display the XA Audio Parameters dialog box if "XA" was selected as the output format in the MovConv main window.

Figure 1-23: XA Audio Parameters Dialog Box

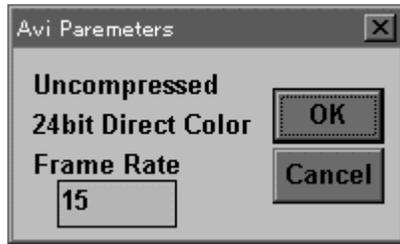


- **Frequency settings**
The output sampling frequency can be selected from the Fs group.
- **Channel settings**
Stereo/Mono settings can be made from the Channel group.

AVI Parameters dialog box

Clicking the [Detail] button will display the AVI Parameters dialog box if "AVI (Uncompressed)" was selected as the output format in the MovConv main window.

Figure 1-24: AVI Parameters Dialog Box

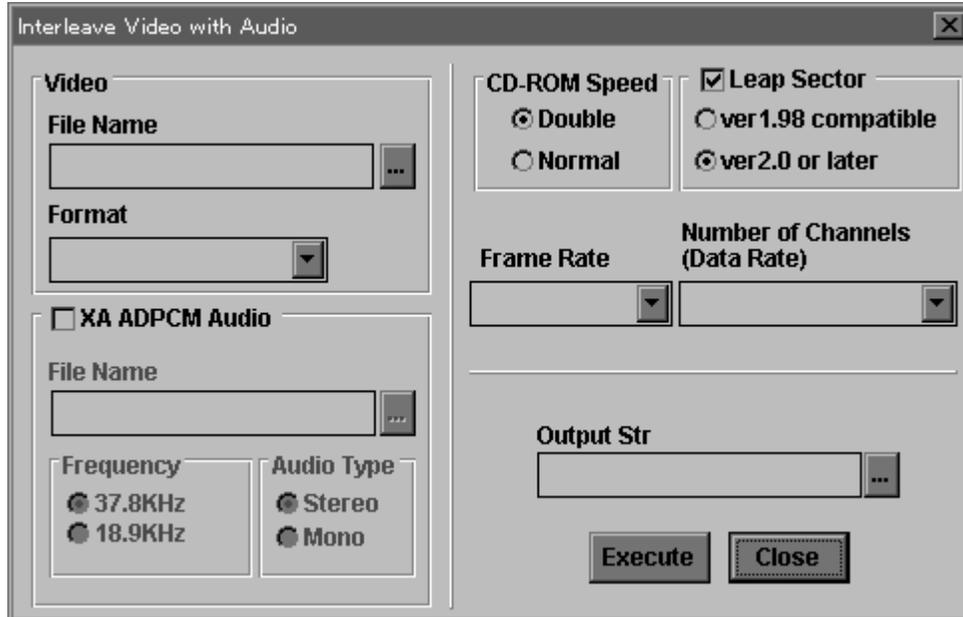


- **Frame rate settings**
The frame rate of the AVI to be output can be entered in the text box. This value must be the same as that for the input data. MovConv does not perform frame rate conversions.

Interleave Video with Audio dialog box

Select the Interleave Video with Audio from the File menu in the MovConv main window to display the Interleave Video with Audio dialog box.

Figure 1-25: Interleave Video with Audio Dialog Box



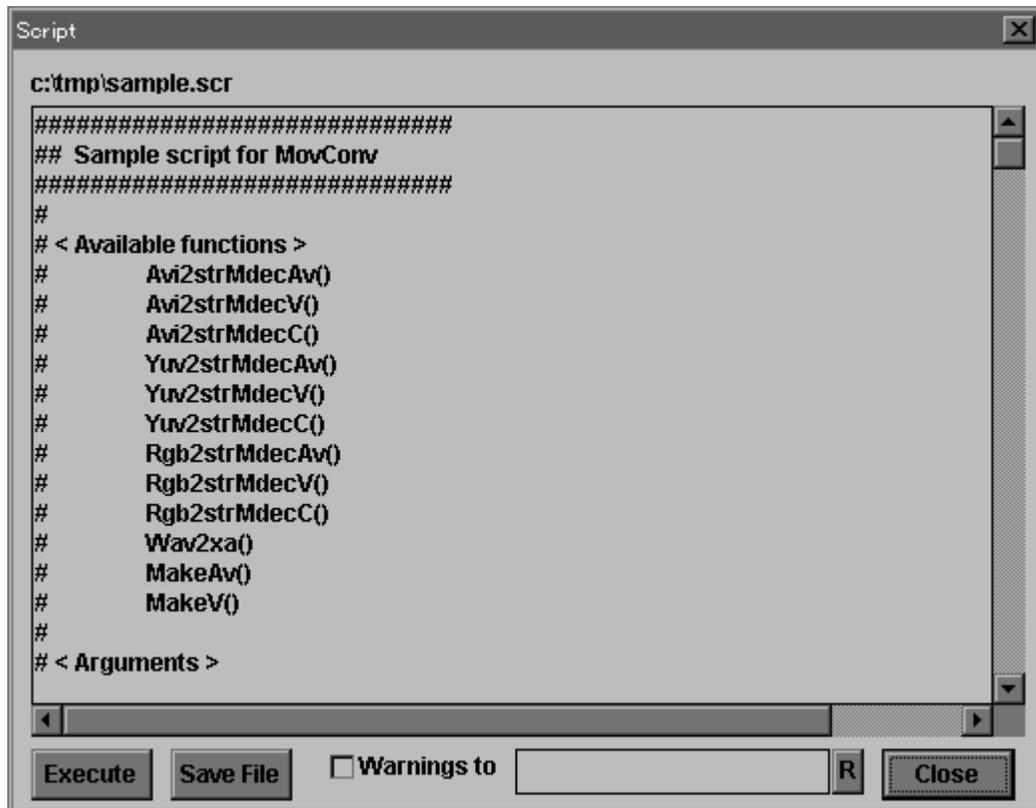
- **Specifying a video filename**
Click the [...] button in the Video group to display the File Open dialog box and select an image file. A filename can be entered directly in the File Name text box as well.
- **Specifying a video format**
Select a format using the Format combo box in the Video group.

- **Set Audio/No audio**
Use the XA ADPCM Audio check box. Check the check box if video and audio are to be interleaved. Leave the box unchecked when performing a simple video format conversion with no audio interleaving. For example, leave the box unchecked when converting a BS sequence to STR(MDEC).
- **Specifying an audio filename**
If Audio is selected, an audio filename can be entered. Click the [...] button in the XA ADPCM Audio group and use the File Open dialog box to select an audio file. A filename can also be entered directly in the FileName text box.
- **Specifying frequency, stereo/mono**
If Audio is selected, a frequency and stereo/mono can be selected. The values for the audio input file must be entered. Interleaving will not be performed correctly if these settings do not match what is in the data.
- **Specifying CD-ROM speed**
Select Double/Normal in the CD-ROM Speed group.
- **Specifying leap sectors**
Check the Leap Sector check box. Refer to the MDEC Parameters dialog box settings to determine if Leap Sector needs to be checked or not.
- **Specifying the leap sector version**
Select ver 1.98 compatible if the video data to be interleaved was created with MovConv version 1.98 or earlier. For data created with version 2.0 or after, select ver 2.0 or later.
- **Specifying a frame rate**
Select a frame rate from the Frame Rate combo box.
- **Number of channels**
Select the number of channels from the Number of Channels(Data Rate) combo box.
- **Specifying an output file**
Click the [...] button from Output Str to display the File Save dialog box and specify a filename. A filename can be entered directly into the Output Str text box as well.
- **Perform interleaving**
Click the [Execute] button to perform interleaving. A gauge serving as a progress indicator will be displayed. Click the [Cancel] button on the gauge to cancel conversion.

The Script dialog box

The Script dialog box is displayed by selecting New Script or Open Script from the File menu in the MovConv main window.

Figure 1-26: Script Dialog Box



- **Editing a script**
The displayed script can be edited directly.
- **Saving a script**
Click the [Save File] button to save the script to a file.
- **Switching destination for message output**
Normally, messages output due to run-time error, etc. are displayed in a dialog box. By checking the Warnings to check box and entering a filename in the text box, messages can be output to the specified file. This prevents operation from stopping and waiting for input during execution of a script, and also provides a log of messages generated during execution.
- **Executing a script**
Click the [Execute] button to execute a script.

Available formats

MovConv can handle the following formats:

AVI (uncompressed)

The AVI Windows video format with uncompressed frame data. The compression format of AVI data can be performed with AVI editing tools.

TIM sequences

A TIM sequence is a series of TIM-format files (the PlayStation static image format) ordered according to the rules shown below. MovConv treats a TIM sequence as a continuous movie.

```
...  
...  
xx0017.TIM  
xx0018.TIM  
xx0019.TIM  
xx0020.TIM
```

```
...  
...
```

(xx is any string, the numbers do not have to be four digits long)

A TIM sequence is specified using the filename that corresponds to the initial frame. The files corresponding to the second and subsequent frames are automatically searched for in the same directory. The entire sequence of TIM files is taken to be one continuous movie.

RGB sequences

RGB is a 24-bit direct color format. Each pixel is represented by 8 bits each of red, green, and blue. Pixels are ordered starting with the pixel in the upper left corner and ending with the pixel in the lower right corner of the screen. There is no width or height specification in RGB format, so the width and height of the image must be entered separately when an RGB file is specified in MovConv. File and filename restrictions are the same as those for TIM sequences.

YUV sequences

In the YUV(422) format, a luminance (Y) value and two chrominance (Cb, Cr) values are arranged in the following manner.

Cb12, Y1, Cr12, Y2, Cb34, Y3, Cr34, Y4

Each component has an 8-bit gradation. Two adjacent pixels will share the same chrominance data so the horizontal resolution for chrominance is 1/2 of the horizontal resolution for luminance. There is no width or height specification in YUV format, so the width and height must be entered separately when a YUV file is specified in MovConv. File and filename restrictions are the same as those for TIM sequences.

STR(MDEC)

In the STR format, frame data is compressed for MDEC. Frame data for individual frames in STR(MDEC) is in the same format as the BS format.

BS sequences

The BS (bit stream) format is a still image format that has been compressed for MDEC. There is no width or height specification in BS format. File and filename restrictions are the same as those for TIM sequences.

WAV

WAV is the Windows audio format. MovConv can handle 16-bit PCM WAV data with sampling frequencies of 44KHz/22KHz/11KHz.

PCM

MovConv can handle 16-bit PCM data.

XA

The XA format supports ADPCM compression according to the CD-ROM XA specification. MovConv can handle stereo/mono data with sampling frequencies of 37.8KHz/19.8KHz.

Scripts

MovConv is able to perform batch processing of multiple operations. The flow of operations is specified in a script. The syntax used in MovConv scripts is very simple and involves specifying a sequence of functions for conversion and interleaving operations. Branching and looping features are not available. The following list shows all the scripting functions defined in MovConv. Function specifications may vary according to the version number of MovConv, so please refer to the sample script file included with MovConv.

Table 1-9: List of Scripting Functions

Function	Action
Avi2strMdecAv()	Create STR(MDEC) from AVI (Easy Settings).
Avi2strMdecV()	Create STR(MDEC) from AVI (Easy Settings).
Avi2strMdecC()	Create STR(MDEC) from AVI (Custom Settings).
Yuv2strMdecAv()	Create STR(MDEC) from YUV (Easy Settings).
Yuv2strMdecV()	Create STR(MDEC) from YUV (Easy Settings).
Yuv2strMdecC()	Create STR(MDEC) from YUV (Custom Settings).
Rgb2strMdecAv()	Create STR(MDEC) from RGB (Easy Settings).
Rgb2strMdecV()	Create STR(MDEC) from RGB (Easy Settings).
Rgb2strMdecC()	Create STR(MDEC) from RGB (Custom Settings).
Wav2xa()	Convert WAV file to XA.
Pcm2xa()	Convert PCM file to XA.
MakeAv()	Interleave video and audio.
MakeV()	Convert video format.
Rgb2bsAV()	Create BS from RGB (Easy Settings).
Rgb2bsV()	Create BS from RGB (Easy Settings).
Rgb2bsC()	Create BS from RGB (Custom Settings).
Yuv2bsAV()	Create BS from YUV (Easy Settings).
Yuv2bsV()	Create BS from YUV (Easy Settings).
Yuv2bsC()	Create BS from YUV (Custom Settings).
Tim2bsAV()	Create BS from TIM (Easy Settings).
Tim2bsV()	Create BS from TIM (Easy Settings).
Tim2bsC()	Create BS from TIM (Custom Settings).
Yuv2avi()	Create AVI from YUV.
Rgb2Avi()	Create AVI from RGB.

Functions ending with "AV" are used when audio interleaving is to be performed later. Functions ending with "V" are used when audio interleaving will not be performed. Functions ending with "C" are used when audio interleaving will not be performed but frame size will be specified. However, in Avi2strMdecAV(), if the source AVI contains audio, the conversion will include the audio, and if the source does not contain audio, it will be converted into data that can be interleaved later.

Avi2strMdecAv(InputFilename, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, AudioFrequency, StereoMono)

```

[Action]          Create STR(MDEC) with audio from AVI
                  (Easy Settings).

[Parameters]

    InputFilename: Specifies the input filename.
    OutputFilename: Specifies the output filename.
    CD-ROMspeed:   Specifies CD-ROM speed. (x1/x2)
    FrameRate:     Specifies the frame rate.
                  (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels: Specifies the number of channels. (1/2/4)
    MDECversion:   Specifies MDEC version number. (2/3)
    LeapSector:    Specifies leap sector setting. (TRUE/FALSE)
    AudioFrequency: Selects XA audio frequency. (37.8KHz/18.9KHz)
    StereoMono:    Indicates Stereo/Mono. (Stereo/Mono)

[Example]

Avi2strMdecAv(
    c:\tmp\data.avi, # Input file name
    c:\tmp\out1.str, # Output file name
    x2,              # CD-ROM speed
    15fps,           # Frame rate
    1,               # Number of channels
    2,               # MDEC version
    FALSE,           # LeapSector
    37.8KHz,        # Frequency of audio
    Stereo           # Stereo or Mono
);

```

Avi2strMdecV(InputFilename, OutputFilename, CD-ROMspeed, FrameRate,NumberOfChannels, MDECversion, LeapSector)

```

[Action]          Create STR(MDEC) without audio from AVI (Easy Settings).

[Parameters]

    InputFilename: Specifies the input filename.
    OutputFilename: Specifies the output filename.
    CD-ROMspeed:   Specifies CD-ROM speed. (x1/x2)
    FrameRate:     Specifies the frame rate.
                  (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels: Specifies the number of channels. (1/2/4)
    MDECversion:   SPECIFIES MDEC VERSION NUMBER. (2/3)
    LeapSector:    Specifies leap sector setting. (TRUE/FALSE)

[Example]

Avi2strMdecV(
    c:\tmp\data.avi, # Input file name
    c:\tmp\out2.str, # Output file name
    x2,              # CD-ROM speed
    15fps,           # Frame rate
    2,               # Number of channels
    2,               # MDEC version
    TRUE             # LeapSector
);

```

Avi2strMdecC(InputFilename, OutputFilename, MaxFrameSize,FixedFrameSize, MDECversion)

```
[Action]                Create STR(MDEC) without audio from AVI
                        (Custom Settings).

[Parameters]

    InputFilename:      Specifies the input filename.
    OutputFilename:     Specifies the output filename.
    MaxFrameSize:       Specifies the maximum size of a frame.
                        ex. '10sectors', '20000bytes'
    FixedFrameSize:     Specifies whether the frame size is fixed or
                        variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:        SPECIFIES MDEC VERSION NUMBER. (2/3)
```

[Example]

```
Avi2strMdecC(
    c:\tmp\data.avi,    # Input file name
    c:\tmp\out3.str,   # Output file name
    10sectors,         # Maximum size for one frame
    FixedFrameSize,    # Fixed frame size or not
    2                  # MDEC version
);
```

Yuv2strMdecAv(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, AudioFrequency, StereoMono)

```
[ Action ]             Create STR(MDEC) with audio from YUV
                        (Easy Settings).

[ Parameters ]

    InputFilename:      Specifies the input filename.
    Width:              Specifies the width of the input image.
    Height:             Specifies the height of the input image.
    OutputFilename:     Specifies the output filename.
    CD-ROMspeed:        Specifies CD-ROM speed. (x1/x2)
    FrameRate:          Specifies the frame rate.
                        (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels:   Specifies the number of channels. (1/2/4)
    MDECversion:        SPECIFIES MDEC VERSION NUMBER. (2/3)
    LeapSector:         Specifies leap sector setting. (TRUE/FALSE)
    AudioFrequency:     SELECTS XA AUDIO FREQUENCY. (37.8KHz/18.9KHz)
    StereoMono:         Indicates Stereo/Mono. (Stereo/Mono)
```

[Example]

```
Yuv2strMdecAv(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out4.str,    # Output file name
    x2,                  # CD-ROM speed
    15fps,              # Frame rate
    1,                   # Number of channels
    2,                   # MDEC version
    TRUE,                # LeapSector
    37.8KHz,            # Frequency of audio
    Mono                 # Stereo or Mono
);
```

Yuv2strMdecV(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector)

[Action] Create STR(MDEC) without audio from YUV (Easy Settings).

[Parameters]

InputFilename: Specifies the input filename.

Width: Specifies the width of the input image.

Height: Specifies the height of the input image.

OutputFilename: Specifies the output filename.

CD-ROMspeed: Specifies CD-ROM speed. (x1/x2)

FrameRate: Specifies the frame rate. (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)

NumberOfChannels: Specifies the number of channels. (1/2/4)

MDECversion: SPECIFIES MDEC VERSION NUMBER. (2/3)

LeapSector: Specifies leap sector setting. (TRUE/FALSE)

[Example]

```
Yuv2strMdecV(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out5.str,    # Output file name
    x2,                 # CD-ROM speed
    15fps,              # Frame rate
    2,                  # Number of channels
    2,                  # MDEC version
    TRUE                # LeapSector
);
```

Yuv2strMdecC(InputFilename, Width, Height, OutputFilename, MaxFrameSize, FixedFrameSize, MDECversion)

[Action] Create STR(MDEC) without audio from YUV (Custom Settings).

[Parameters]

InputFilename: Specifies the input filename.

Width: Specifies the width of the input image.

Height: Specifies the height of the input image.

OutputFilename: Specifies the output filename.

MaxFrameSize: Specifies the maximum size of a frame. ex. '10sectors', '20000bytes'

FixedFrameSize: Specifies whether the frame size is fixed or variable. ('FixedFrameSize/VariableFrameSize')

MDECversion: SPECIFIES MDEC VERSION NUMBER. (2/3)

[Example]

```
Yuv2strMdecC(
    c:\tmp\data0000.yuv, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out6.str,    # Output file name
    10sectors,          # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                    # MDEC version
);
```

Rgb2strMdecAv(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, AudioFrequency, StereoMono)

```
[ Action ]                Create STR(MDEC) with audio from RGB
                           (Easy Settings).

[ Parameters ]

  InputFilename:          Specifies the input filename.
  Width:                  Specifies the width of the input image.
  Height:                 Specifies the height of the input image.
  OutputFilename:         Specifies the output filename.
  CD-ROMspeed:            Specifies CD-ROM speed. (x1/x2)
  FrameRate:              Specifies the frame rate.
                           (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
  NumberOfChannels:       Specifies the number of channels. (1/2/4)
  MDECversion:            SPECIFIES MDEC VERSION NUMBER. (2/3)
  LeapSector:             Specifies leap sector setting. (TRUE/FALSE)
  AudioFrequency:        SELECTS XA AUDIO FREQUENCY. (37.8KHz/18.9KHz)
  StereoMono:             Indicates Stereo/Mono. (Stereo/Mono)
```

[Example]

```
Rgb2strMdecAv(
  c:\tmp\data0000.rgb, # Input file name
  320,                 # Width
  240,                 # Height
  c:\tmp\out7.str,    # Output file name
  x2,                  # CD-ROM speed
  15fps,               # Frame rate
  1,                   # Number of channels
  2,                   # MDEC version
  TRUE,                # LeapSector
  37.8KHz,             # Frequency of audio
  Mono                 # Stereo or Mono
);
```

Rgb2strMdecV(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector)

```
[Action]                  Create STR(MDEC) without audio from RGB
                           (Easy Settings).

[Parameters]

  InputFilename:          Specifies the input filename.
  Width:                  Specifies the width of the input image.
  Height:                 Specifies the height of the input image.
  OutputFilename:         Specifies the output filename.
  CD-ROMspeed:            Specifies CD-ROM speed. (x1/x2)
  FrameRate:              Specifies the frame rate.
                           (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
  NumberOfChannels:       Specifies the number of channels. (1/2/4)
  MDECversion:            SPECIFIES MDEC VERSION NUMBER. (2/3)
  LeapSector:             Specifies leap sector setting. (TRUE/FALSE)
```

```
[ Example ]
Rgb2strMdecV(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out8.str,    # Output file name
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    2,                   # Number of channels
    2,                   # MDEC version
    TRUE                 # LeapSector
);
```

Rgb2strMdecC(InputFilename, Width, Height, OutputFilename, MaxFrameSize, FixedFrameSize, MDECversion)

```
[ Action ]                Create STR(MDEC) without audio from RGB
                          (Custom Settings).

[ Parameters ]

    InputFilename:        Specifies the input filename.
    Width:                Specifies the width of the input image.
    Height:               Specifies the height of the input image.
    OutputFilename:       Specifies the output filename.
    MaxFrameSize:         Specifies the maximum size of a frame.
                          ex. '10sectors', '20000bytes'
    FixedFrameSize:       Specifies whether the frame size is fixed or
                          variable. ('FixedFrameSize/VariableFrameSize)
    MDECversion:          SPECIFIES MDEC VERSION NUMBER. (2/3)

[ Example ]
Rgb2strMdecC(
    c:\tmp\data0000.rgb, # Input file name
    320,                 # Width
    240,                 # Height
    c:\tmp\out9.str,    # Output file name
    10sectors,           # Maximum size for one frame
    FixedFrameSize,     # Fixed frame size or not
    2                     # MDEC version
);
```

Wav2xa(InputFilename, OutputFilename, AudioFrequency, StereoMono)

```
[Action]                  Convert WAV file to XA.

[Parameters]

    InputFilename:        Specifies the input filename.
    OutputFilename:       Specifies the output filename.
    AudioFrequency:       SELECTS XA AUDIO FREQUENCY. (37.8KHz/18.9KHz)
    StereoMono:           Indicates Stereo/Mono. (Stereo/Mono)

[Example]
Wav2xa(
    c:\tmp\data.wav,     # Input file name
    c:\tmp\data.xa,     # Output file name
    37.8KHz,             # Frequency of xa audio
    Stereo                # Stereo or Mono for xa audio
);
```

Pcm2xa(InputFilename, PCMbitsPerSample, PCMfrequency, PCMstereoMono, OutputFilename, XAfrequency, XAstereoMono)

```

[Action]                Convert PCM file to XA.

[Parameters]

    InputFilename:      Specifies the input filename.
    PCMbitsPerSample:   Specifies the number of sampling bits in
                        PCM data. (16bit)
    PCMfrequency:       Specifies frequency of PCM data (in KHz).
    PCMstereoMono:      Specifies stereo/mono of PCM data. (Stereo/Mono)
    OutputFilename:     Specifies the output filename.
    XAfrequency:        SELECTS XA AUDIO FREQUENCY. (37.8KHz/18.9KHz)
    XAstereoMono:       Indicates Stereo/Mono. (Stereo/Mono)

[Example]

Pcm2xa (
    c:\tmp\data.pcm,    # Input file name
    16bit,              # Bits Per Sample
    44.1KHz,           # Frequency of PCM audio (KHz)
    Stereo,            # Stereo or Mono for PCM audio
    c:\tmp\data.xa,    # Output file name
    37.8KHz,           # Frequency of xa audio
    Stereo             # Stereo or Mono for xa audio
);

```

MakeAv(VideoFilename, VideoFormatName, AudioFileName, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, LeapSector, OldLeapSector, AudioFrequency, StereoMono)

```

[Action]                Interleave video and audio.

[Parameters]

    VideoFilename:      Specifies the input filename.
    VideoFormatName:    Specifies the video format name. (strMDEC/bs)
    AudioFileName:       Specifies the audio filename.
    OutputFilename:     Specifies the output filename.
    CD-ROMspeed:        Specifies CD-ROM speed. (x1/x2)
    FrameRate:          Specifies the frame rate.
                        (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
    NumberOfChannels:    Specifies the number of channels. (1/2/4)
    LeapSector:         Specifies leap sector setting. (TRUE/FALSE)
    OldLeapSector:      Indicates whether the leap sector version is old.
                        TRUE if the video was created with MovConv
                        version 2.0 or before. Otherwise False.
                        (TRUE/FALSE)
    AudioFrequency:     SELECTS XA AUDIO FREQUENCY. (37.8KHz/18.9KHz)
    StereoMono:         Indicates Stereo/Mono. (Stereo/Mono)

```

[Example]

```
MakeAv (
    c:\tmp\data.str,      # Video file name
    strMDEC,             # Video format name
    c:\tmp\data.xa,      # Audio file name
    c:\tmp\out10.str,    # Output file name (audio & video)
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    1,                   # Number of channels
    FALSE,               # LeapSector
    FALSE,               # ver1.98 compatible leap sector
    37.8KHz,             # Frequency of audio
    Stereo                # Stereo or Mono
);
```

MakeV(VideoFilename, VideoFormatName, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, LeapSector, OldLeapSector)

[Action] Convert video format.

[Parameters]

VideoFilename:	Specifies the input filename.
VideoFormatName:	Specifies the image format name. (strMDEC/bs)
OutputFilename:	Specifies the output filename.
CD-ROMspeed:	Specifies CD-ROM speed. (x1/x2)
FrameRate:	Specifies the frame rate. (10fps/12.5fps/15fps/16.7fps/20fps/25fps/30fps)
NumberOfChannels:	Specifies the number of channels. (1/2/4)
LeapSector:	Specifies leap sector setting. (TRUE/FALSE)
OldLeapSector:	Indicates whether the leap sector version is old. TRUE if the video was created with MovConv version 2.0 or before. Otherwise False. (TRUE/FALSE)

[Example]

```
MakeV (
    c:\tmp\data.str,      # Video file name
    strMDEC,             # Video format name
    c:\tmp\out11.str,    # Output file name (video)
    x2,                  # CD-ROM speed
    15fps,               # Frame rate
    1,                   # Number of channels
    TRUE,                # LeapSector
    FALSE                 # ver1.98 compatible leap sector
);
```

Restrictions

- MovConv only handles 16-bit WAV and PCM audio data.
- Previewing must be performed on displays with a bit depth of 16-bits or greater.

MovPack: Packs multiple streaming/audio data sets into single pack

Version 1.6

Documentation date: 8/6/97

Description

MovPack is a tool for consolidating (packing) multiple sets of streaming data or CD-ROM XA audio data into a single package. MovPack can be used in the following situations:

- When creating multi-channel streaming data
- When recording CD-ROM XA audio data to a CD-ROM

For details on multi-channel streaming, please refer to the Movie Converter Tutorial.

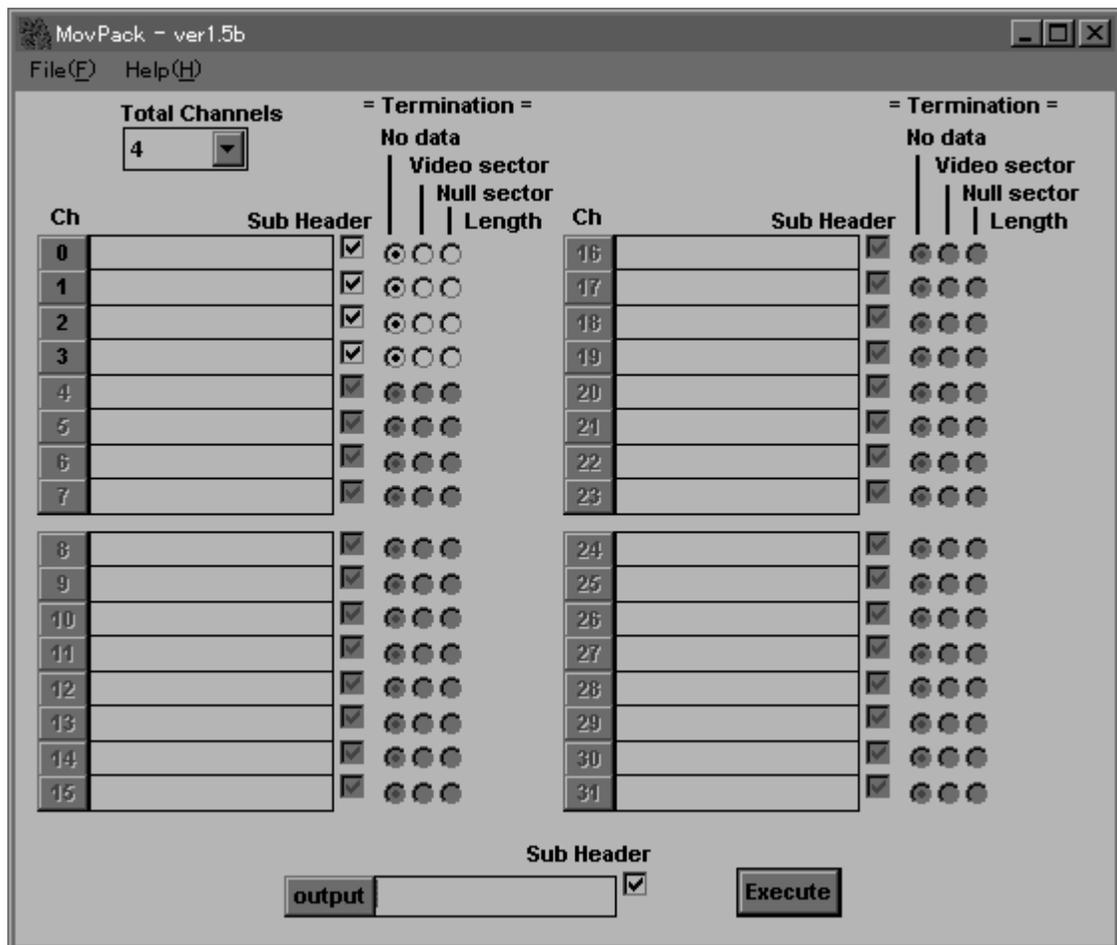
How to use

Please refer to the Movie Converter Tutorial for examples and specific information on usage. This section will describe how to enter settings in the MovPack dialog box.

The MovPack main window

The MovPack main window is displayed when MovPack is started.

Figure 1-27: MovPack



- **Specifying the total number of channels**

The number of total channels is entered in the Total Channels combo box. The table below shows the settings that are entered if the data to be interleaved is XA audio data.

Table 1-10: Settings for the total number of channels in XA audio data

Playback speed	Sampling frequency	Stereo/Mono	Total number of channels
2 X speed	37.8KHz	Stereo	8
2 X speed	37.8KHz	Mono	16
2 X speed	18.9KHz	Stereo	16
2 X speed	18.9KHz	Mono	32
1 X speed	37.8KHz	Stereo	4
1 X speed	37.8KHz	Mono	8
1 X speed	18.9KHz	Stereo	8
1 X speed	18.9KHz	Mono	16

The XA data output from MovConv will not play back properly if recorded directly to the CD-ROM. For correct playback, the data must be interleaved according to the number of channels, as shown in the table.

- **Specifying the input filename**

For each channel, an input filename must be specified. Clicking on a channel button will open up a File Open dialog box. Select an input file here and click the "OK" button. A filename can also be entered directly in the text box. If the number of files to be interleaved is less than the total number of channels, the unused entries can be left blank. If no filename is specified, the channel will be left empty. For multi-channel streaming data, select STR(MDEC) for the input files. For audio data, select XA for the input files.

- **Setting input data sub header flags**

A subheader flag is set for channels for which files are specified. Check the Sub Header check box if the input data is streaming audio data or audio data. Leave the box unchecked if the streaming data is image-only.

- **Specifying termination data**

Termination data to be added to the end of the channel data is specified. Normally, "No data" should be selected. Selecting "Video sector" will add empty video sector data at the end of the channel data. This setting can be made if, for example, audio data needs to be padded. By adding video sectors to the end of audio data, the end of audio data can be detected by testing for video sector reads during playback. If "Null sector" is selected, null sectors will be added to the end of the channel data. This setting can be made if a gap needs to be added at the end of the data. If Video sectors or Null sectors are specified for the termination data, the length in sectors must also be entered.

- **Specifying an output filename**

Clicking the [output] button will open up a File Save dialog box. Select the output filename here and click the [OK] button. A filename can also be entered directly in the text box.

- **Specifying an output data sub header flag**

Check the Sub Header check box if the output data is to be streaming data with audio, or audio data. Leave the check box unchecked if the streaming data is image-only.

- **Interleaving**

Click the [Execute] button to begin interleaving.

Chapter 2:

3D Graphic Conversion Utilities

DXF2RSD: Converts DXF to RSD

Version 2.81

Documentation date: 4/25/96

Description

This software converts the DXF file into the PlayStation 3D model data format (RSD).

How to use

DXF2RSD [options] DXF-files...

Input of a DXF file with the argument creates 4 files shown below.

- File-related information file (*.RSD)
- Polygon information file (*.PLY)
- Material information file (*.MAT)
- Group information file (*.GRP)

These 4 files are put together into one and called "RSD" or "RSD data". The wildcard is available for the argument. You can input and convert multiple files at a time. Omission of ".DXF" of the filename is allowed.

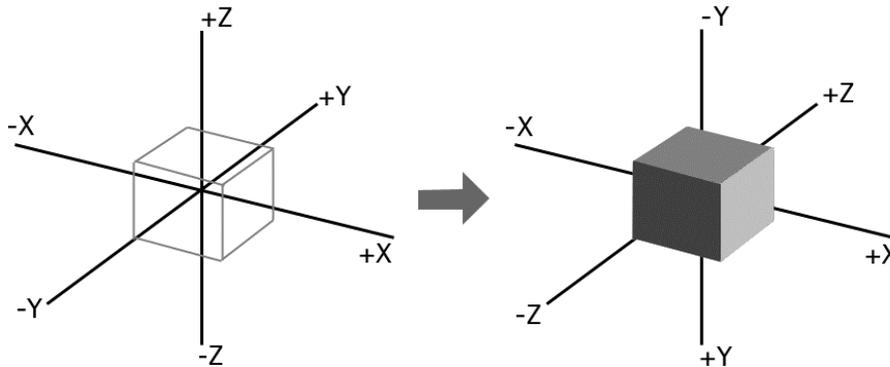
Options

<i>-o output-file</i>	Specify an output RSD name. The extension is removed. The default creates a filename identical with the input filename, except the extension which is removed.
<i>-col r g b</i>	Specify a color of the whole model with R, G and B (0-255, respectively). The default is the gray (200 200 200).
<i>-cf color-file</i>	Specify a color table file.
<i>-cl</i>	Outputs a list of undefined colors to a standard output. This sorts polygons of the same color and outputs them to a MAT file. The default is OFF. (Refer to example 2.)
<i>-info</i>	Shows information about the input DXF file. This lets you know the approximate size and number of polygons. No conversion is executed.
<i>-max n-poly</i>	Set the maximum number of polygons which can be converted. The default is 10000.
<i>quad or -quad1</i>	Triangulates no 4-vertex 3DFACE. This allows reduction of the number of polygons of the whole model. The default is OFF.
<i>-quad2 (threshold)</i>	Synthesizes 2 adjacent triangles into one quadrangle. When an argument is designated, the movement of the direction (normal vector) of the two triangles can be set by the angle and triangles below that angle are synthesized. When the argument is 0.0 synthesis takes place only when it has the same normal vector, and when the argument is 10.0, a differential of up to 10 degrees is acceptable. Designate the fractions between 0.0 and 90.0. The default is 1.0. (Refer to example 4).
<i>-quad3</i>	Converts the triangle into the quadrangle whose third and fourth vertices are equal to each other. This allows any triangle to turn to a quadrangle.
<i>-s or -g</i>	Executes the smooth (Gouraud) shading. The Default is OFF.
<i>-e distance</i>	Regards all vertices which exist in a sphere having a radius of distance given with the argument as the same. This allows reduction of the numbers of vertices and polygons. Distance calculation is executed after scale-up or scale-down with the -sc option.

2-4 3D Graphic Conversion Utilities

- r* Prevents overlapping normal vectors from being created to reduce the number of normals. This is useful to the flat shading. The default is OFF. (Refer to example 3.)
- n* Creates no normal. Set this when no light calculation is required. The default is OFF.
- sc factor* Scales up or down the model. Specify a scale factor with the argument. The default is 1.0.
- t x y z* Moves the model. Specify a movement with the argument. The default is (0.0, 0.0, 0.0).
- auto* Moves the model near to the origin and scales up or down it (so that it can settle in a cube having a side of 1000.) The default is OFF.
- back* Reverses the normal vector of all polygons. The default is OFF.
- both* Creates both sided polygons. The default is OFF.
- dup* Creates both front and rear polygons of all polygons. This doubles the number of polygons. The default is OFF.
- nopl* Converts only the 3DFACE, ignoring Polyline. The default is OFF.
- Y-Z, -Y+Z, +Y-Z, +Y+Z, -Z-Y, -Z+Y, +Z-Y, +Z+Y*
Specify a coordinate system conversion method. Specify the front and upper axes and their directions of a coordinate system of a modeler which is seen from the front. For example, "-Y+Z" means that the front axis is -Y axis and the upper axis is +Z axis. The coordinate system which is mentioned here is for the DXF and does not always correspond to the modeler screen. The default is "-Y+Z". DXF2RSD converts it into the PlayStation coordinate system (-Z-Y). (In the PlayStation coordinate system, the front axis is -Z axis and the upper axis is -Y axis.)

Figure 2-1: Conversion of Coordinate Systems



Coordinate system of modeler (-Y+Z type) Coordinate system of PlayStation (-Z-Y type)

- v* Outputs detailed information about conversion to a standard output. (Refer to example 1 shown later.)

Main improvements from previous version (ver. 2.8)

The memory bug has been removed. (DOS)>

An SGI version has been added.

Constraint

The current version has the following constraints:

1. Is compatible with only the ASCII format DXF.
2. Is compatible with only the 3DFACE and Polyline of DXF entities.
3. May not convert a large Polyline. Convert it to the 3DFACE (triangle or quadrangle) as many as possible with a modeler and then create the DXF file.
4. The current -quads [1-3] do not couple 2 triangles to create a quadrangle. Quadrangle which does not have 4 vertices on the same plane may not be displayed correctly.
5. Numbers of vertices and normals to be created affect the maximum number of polygons which can be converted. Generally, it is about 5000 polygons.

Supplement

Both the 3DFACE and Polyline are DXF polygon representation formats. The 3DFACE represents a polygon (triangle or quadrangle) with 4 vertices. On the other hand, the Polyline represents multiple polygons with coupled segments. The 3DFACE is apt to cause data to be larger but has the superior compatibility. The Polyline allows data to be smaller but data conversion between different modelers may not be realized correctly.

The 3DFACE can be converted directly into the RSD but the Polyline requires triangulation before conversion. The triangulation costs much and is not always successful. (An error message "Fail to triangulate!" appears.) Even if the triangulation is completed successfully, directions of some polygons may be reversed.

However, a Polyline which 3D studio and others create is called POLYFACE MESH and equal to the 3DFACE representation. Therefore, there is no problem about conversion.

A practical number of polygons which can be moved as an object on PlayStation is about 2000 at maximum. Create a model data, taking into consideration of it.

Selection of the Gouraud shading may allow conversion even if conversion with the flat shading is impossible because of too many polygons.

Every exchange of Y and Z coordinate axes or of plus and minus signs of each axis reverses the front and back of the polygon.

For some modelers, even the 3DFACE output may reverse the front and back of the polygon. If the whole polygon is reversed, change the coordinate system (for example, +Y+Z) or use the - back option. If it is reversed partly, reversion of the part with the modeler or material editor is required.

Commercial 3D modelers which are found to be convertible with DXF2RSD are shown below. Options for coordinate system conversion are also given.

Table 2-1: Commercial 3D modelers convertible with DXF2RSD

3D modeler (version)	Platform	Coordinate conversion option
form Z (2.1 and later)	Mac	-Y+Z
Strata Studio Pro (1.0)	Mac	+Z+Y
StrataVision	Mac	+Z+Y
Sculptura (1.1)	PC	-Y+Z
trueSpace (1.0)	PC	-Y-Z
Alias Upfront (1.1)	PC	-Y+Z
3D Studio	PC	-back

Commercial 3D modelers which are found to be unconvertible now are as follows.

Table 2-2: Currently unconvertible commercial 3D modelers

MacroModel (1.5)	PC
ShadeIII (1.0)	Mac

The sufficient condition for "To be convertible" is that it can output the 3DFACE DXF. Some are convertible even if they do not satisfy the condition. Moreover, some which cannot be converted directly may become convertible when you read in and save the DXF once with another modeler.

This software creates a RSD without normal, regarding that the -n option is specified unless a too large data allows creation of all normals.

The -r option is invalid in the Gouraud shading. You cannot use the -r option in the normal MIMe animation because the number and order of normals are changed if the -r option is specified.

When synthesizing two triangles into a quadrangle in -quad2 Z Sort may occur.

When the two triangles designated and synthesized in -quad2 have each been specified as different colors by the modeler and output as DXF, that section alone can invalidate -quad2. In such cases, also simultaneously set the -cl option as with dxf2rsd -quad2 -cl.

Examples

Example 1: Output when the -v option is used

```
C:> DXF2RSD -v -auto +Z+Y -quad -s FOO
```

Input DXF file: FOO.dxf	input DXF File name
[DXF] SIZE : 40230 lines	number of lines in DXF
VERTEX : 4320	number of vertices in DXF
POLYGON : 1468 (estimate)	estimated number of polygons
3-poly : 1376	(content) triangular configurations 1376
4-poly : 32	quadrilateral configurations 32
(9<)-poly : 2	configurations of decagon or greater 2
polylines : 2 (max size=32)	polyline 2 (32 vertices)
RANGE x : -1.015 ... +0.785	minimum and maximum value of each axis
y : -2.533 ... +0.768	Y,Z axes are converted to PlayStation
z : -1.161 ... +0.625	as required
SCALE : 302.870	scaling factor
MOVE : (dx,dy,dz)=(34.788,267.255,81.207)	movement
MATERIAL : 0	number of colored polygons
[RSD] VERTEX : 796	number of vertices after conversion
POLYGON : 1468	number of polygons after conversion
triangles : 1436	(content) triangular configurations 1436
quadrangles : 32	quadrilateral configurations 32
RANGE x : -272.477 ... +272.477	resizing minimum and maximum
y : -500.000 ... +500.000	value of each axis after move
z : -270.510 ... +270.510	
MATERIAL : 0	number of polygons with material
NORMAL : 796	number of normal lines
Output files : FOO.[rsd, ply, mat, grp]	output file name

Example 2: Use of color information

You can reflect color information you entered with a modeler on the RSD. Specify the -cl option. Polygons will be "reasonably" colored. The word "reasonably" means that a color which may differ from the color on the modeler is allocated to parts which had the same color. This is because the DXF holds only "color number" but the RGB value. Then, you can specify a color by editing the MAT file with a text editor or creating a color table file as follows.

```
C:> DXF2RSD -cl FOO > FOO.CL
C:> TYPE FOO.CL
    183
    40
    253
    0
    8
```

Enter a color which you want to allocate to each color number just after the color number with R, G and B values (0-255, respectively) by the text editor.

```
C:> TYPE FOO.CL
    183 100 100 200
    40  58  20  43
    253 10  100 10
    0   212 20  100
    8   0   128 126
```

(As material files, FOO.MAT which are sorted by color number have been created, you may edit directly the RGB field of the file. For MAT file specification, refer to "Appendix: File format RSD" or RSD.TXT.)

Specify the -cf option to start up DXF2RSD again.

```
C:> DXF2RSD -cf FOO.CL FOO
```

Colors which were specified in the color table file, FOO.CL are allocated to a newly created RSD. (The FOO.MAT is updated.)

(If you edit directly the FOO.MAT, restart-up of DXF2RSD is not required.)

Example 3: Conversion of large data

When a too detailed data exists, you can use the -e option to put together several vertices into one. In the following example, numbers of vertices, polygons and normals are reduced by regarding 2 vertices which are less 100 away from each other as a vertex. (Note that distance calculation is based on a scale after scale-up.)

Though depending on data, you can reduce the quantity of data without changing practically the form if you set a proper distance.

```
C:> DXF2RSD -v -e 100 -sc 1000 BIG.DXF
```

Output	Description
Input DXF file : BIG.DXF	
[DXF] SIZE	: 134628 lines
VERTEX	: 18982
POLYGON	: 8618 (estimate)
3-poly	: 1746
4-poly	: 3436
RANGE x	: -1.644 ... +1.545
y	: -2.352 ... +0.000
z	: -3.649 ... +3.993
SCALE	: 1000.000
MATERIAL	: 0
[RSD] VERTEX	: 1208
POLYGON	: 2708 (68% reduced)
triangles	: 2708
RANGE x	: -1642.811 ... +1545.072
y	: -2352.365 ... +0.000
z	: -3649.154 ... +3992.687
MATERIAL	: 0
NORMA	: 2708
Output files : BIG.[rsd, ply, mat, grp]	

Moreover, you can reduce the number of normals if you set the -r option.

Example 4: Quadrangle synthesis

Two adjoining triangles can be synthesized into a quadrangle when setting the -quad2 option. In the following example, by converting an angle of less than 5 degrees between the two triangles into a quadrangle, the number of polygons is reduced by almost half.

```
>DXF2RSD -V -QUAD2 5.0 EARTH

Input DXF file      : earth.dxf
[DXF] SIZE         : 88158 lines
VERTEX             : 8811
POLYGON            : 2937 (Estimate)
                   3-poly : 2937 <-originally 2937 triangles
RANGE  x           : -4.000 ... +3.986
       y           : -3.997 ... +3.997
       z           : -4.000 ... +4.000
MATERIAL:          0

[RSD] VERTEX       : 2231
      POLYGON      : 1686 <-2502 triangles are synthesized
                  triangles      : 43          Became 1251 quadrangles
                  quadrangles    : 1251       Total of 1686 polygons
MATERIAL:          0
NORMAL: 1686 (quad2<4.9870)

Output files : earth. [rsd,ply,mat,grp]
```

The “4.9870” in NORMAL: 1686 (quad2 < 4.9870) in the middle of the figure actually refers to the largest converted angle within the approved angles (in this case 5.0). Therefore, when “-quad2 4.986” is specified and executed for a second time, the number of synthesized quadrangles decreases slightly.

DXF2RSDW: Converts DXF to RSD (Windows version)

Version 1.20

Documentation date: 10/16/95

Description

DXF2RSDW.EXE is a Windows application which allows you to convert the DXF format file which various modelers output into the PlayStation RSD format. This utility has the same capabilities as the dxf2rsd command line version. Execute it by double clicking on the icon registered in either the file manager or program manager. Since dxf2rsdw help can be displayed in the help menu, please refer to this for details.

Options

None

Notes

This current version is the command line version of dxf2rsd with GUI added. As a result, the restrictions, etc. which are present in dxf2rsd are also present in dxf2rsdw.

Messages such as 'now converting', 'conversion complete', etc. are displayed in the status bar located in the lowest section of the application window. When conversion cannot be carried out properly, please check the message in the status bar. Ways of dealing with error messages are explained in the command line version of the dxf2rsd, so by adjusting the parameters based on that information, conversion is made possible.

Changes in version 1.05

Undefined color files can now be output correctly.

The maximum number of polygons which can be handled has been greatly increased. Along with this, the default for the maximum number of polygons has been changed from 2500 to 10000.

Changes in version 1.10

The user interface has been changed. For details on this, please refer to the on-line help (dxf2rsdw.hlp).

Changes in version 1.20

The 'synthesized square' added in dxf2rsd ver2.8 as a function to handle the quad2 option has been added to the 'polygon shapes' selections.

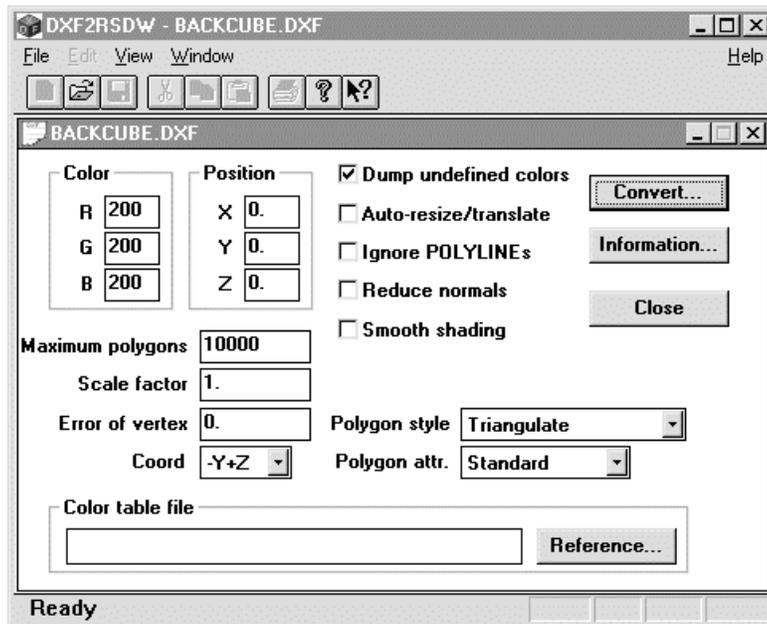
When the 'undefined color display' option is selected, a color coded group is created and written to the GRP file.

How to use

Selection of a DXF file with "Open..." option of the "File" menu lets you access the Parameter Setting window. Change properly parameter values and then press the "Convert..." button or select "Give name to save..." option from the filename. The File Save dialog pops up. Entering a post-conversion filename starts format conversion.

You can drag an input file from the File manager to drop it in the DXF2RSDW.EXT window instead of using the "Open..." command.

Figure 2-2: Information Dialog



- Explanation of Parameter Setting window settings**
 Options corresponding to DXF2RSD.EXE of the Commandline version are also mentioned.
- Color of the whole**
 Specify a color of the whole model with R, G and B values (0-255, respectively). This corresponds to the `-col r g b` option of the Commandline version.
- Model location**
 Moves the model. Specify a movement. This corresponds to the `-t x y z` option of the Commandline version.
- Maximum number of polygons**
 Set the maximum number of polygons which can be converted.

 This corresponds to the `-max n-poly` option of the Commandline version.
- Scale factor**
 Scales up or down the model. Specify a scale factor with a real number. This corresponds to the `-sc factor` of the Commandline version.
- Minimum vertex distance**
 Regards all vertices which exist in a sphere having a radius of distance given here as the same. This allows reduction of the numbers of vertices and polygons. Distance calculation is executed after scale-up or scale-down with a scale factor. This corresponds to the `-distance` option of the Commandline version.

- **Coordinate system**
Specify a coordinate system conversion method. Specify the front and upper axes and their directions of the coordinate system which is seen from the front. For example, "-Y+Z" means that the front axis is -Y axis and the upper axis is +Z axis. The coordinate system which is mentioned here is for the DXF and does not always correspond to the modeler screen. DXF2RSD converts it into the PlayStation coordinate system (-Z-Y). (In the PlayStation coordinate system, the front axis is -Z axis and the upper axis is -Y axis.)

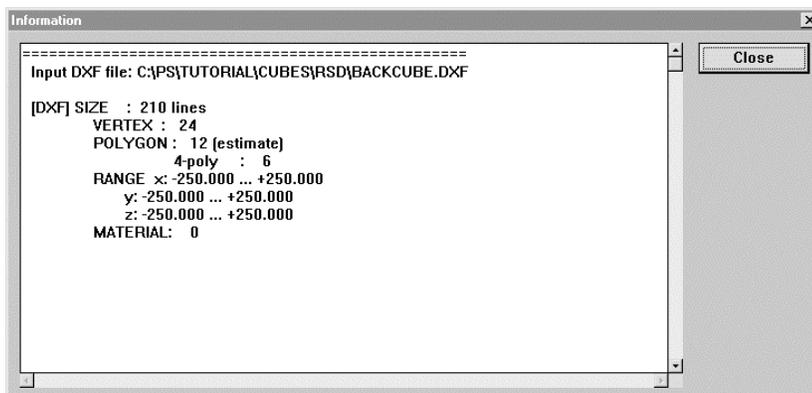
This corresponds to various coordinate system options of the Commandline version.
- **Display of undefined colors**
This corresponds to the -cl option of the Commandline version.

The default of this option differs from that of the Commandline version and is ON.
- **Automatic size adjustment**
Moves the model near to the origin and scales up or down it (so that it can settle in a cube having a side of 1000.)

This corresponds to the -auto option of the Commandline version.
- **Polyline ignore**
Ignores the Polyline and converts only the 3DFACE. This corresponds to the -nopl option of the Commandline version.
- **Normal reduction**
Prevents overlapping normal vectors being from created to reduce the number of normals. This is particularly useful to the flat shading. This corresponds to the -r option of the Commandline version.
- **Gouraud shading**
Executes the smooth (Gouraud) shading. This corresponds to the -s or -g option of the Commandline version.
- **Various polygons**
There are options shown below:
 - **Triangle**
Triangulates all polygons.
 - **Triangulation prohibit**
Triangulates no 4-vertex 3 DFACE. This allows reduction of the number of polygons of the whole model. This corresponds to the -quad option of the Commandline version.
 - **Quadrangle**
Converts the triangle into the quadrangle whose third and fourth vertices are equal to each other. This corresponds to the -quad3 option of the Commandline version.
- **Polygon attributes**
There are options shown below:
 - **Standard**
Creates polygons and normals according to information in the DXF file.
 - **Normal reverse**
Reverses normal vectors of all polygons. This corresponds to the -back option of the Commandline version.
 - **Both sided polygon**
Converts all polygons into both sided polygons. This corresponds to the -both option of the Commandline version.
 - **Duplicate polygons**
Creates both front and rear polygons of each polygon. This corresponds to the -dup option of the Commandline version.

- **Create no normal**
Creates no normal. Specify this when no light calculation is required. This corresponds to the -n option of the Commandline version.
- **Color table file**
Specify a color table file. This corresponds to the -cf color-file option of the Commandline version.
Pressing the "Refer..." button lets the File dialog pop up.
This allows you to enter a filename.
- **Conversion**
Executes format conversion according to specified parameters. As the File Save dialog pops up, specify a post-conversion filename.
This function is equal to "Give name to save..." option of the file menu.
- **Information**
Shows information about the input DXF file. You can know the approximate size and number of polygons.
This corresponds to the -info option of the Commandline version.
They are shown in a dialog shown below.

Figure 2-3: Information Dialog



Selection of the text area with the mouse and pressing Ctrl + C copies them to the Windows clip board. To save them in a file and others, paste them with a proper text editor or others.

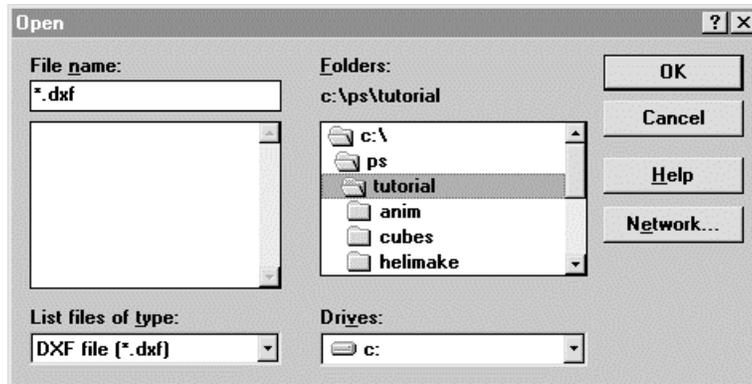
- **Close**
Closes the current window. This function is equal to "Close" option of the file menu.

Menu bar

"Open..." command ([File] menu)

Opens existing DXF files. The following dialog box appears.

Figure 2-4: File Open Dialog



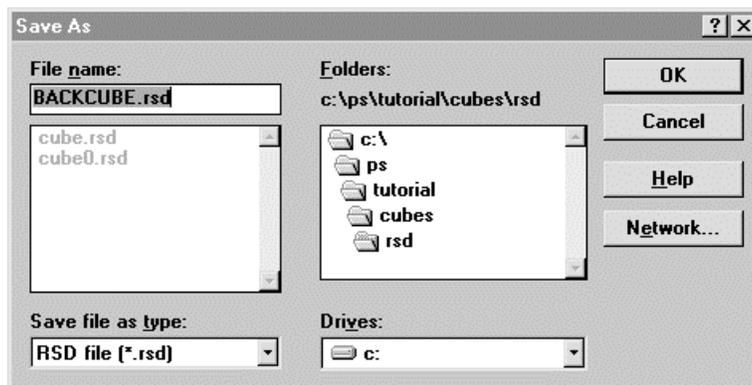
This dialog allows you to set the following options.

- **File name**
Enter or select a DXF filename you want to open. The List box shows files which have an extension corresponding to a file format you selected in the "File type" box.
- **File type**
Set whether you want to filter the file list with the *.dxf pattern or not.
- **Drive**
Select a drive in which a DXF file you want to open is saved.
- **Directory**
Select a directory in which a DXF you want to open is saved.

"Give name to save..." command ([File] menu)

Give a filename to a DXF file you are dealing with to save it. As the following dialog box appears, you can give a filename you want to a DXF file you are dealing with to save it. This executes format conversion to the RSD format according to parameters you set.

Figure 2-5: File Save Dialog



This dialog box allows you to set the following options to specify a filename or location in which you save it.

- **Filename**
Enter a filename different from the original name. Give a filename of up to 8 half size characters and add an extension of 3 half size characters to it.
- **Drive**
Select a drive in which you save the RSD file.
- **Directory**
Select a directory in which you save the RSD file.

Constraints

The current version has the following constraints:

Is compatible with only the ASCII format DXF file.

Is compatible with only the 3DFACE and Polyline of the DXF entities.

May not convert a large POLYLINE. Convert it to the 3DFACE (triangle or quadrangle) with a modeler as many as possible and then create the DXF file.

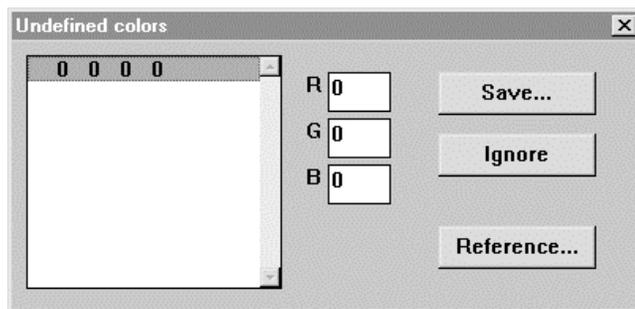
"Triangulation prohibit" never couples 2 triangles to create a quadrangle. Quadrangle which does not have 4 vertices on the same plane may not be displayed correctly.

Numbers of vertices and normals to be created affect the maximum number of polygons which can be converted. Generally, it is about 5000 polygons.

Use of color information

You can reflect color information you entered with the modeler on the RSD. Specify "Display undefined colors" option. Polygons will be "reasonably" colored. The word "reasonably" means that a color which may differ from the color on the modeler is allocated to parts which had the same color. This is because the DXF holds only "color number" but the RGB value. Then, you can edit the MAT file with a text editor or specify a color by creating a color table file with the "Display undefined colors" option. In this case, "Undefined color" dialog pops up after completion of conversion from the DXF to the RSD.

Figure 2-6: Undefined Color Dialog



This dialog shows a list of undefined colors. Edit R, G and B fields and then press the "Save..." button. You can give a proper filename with the File dialog to save the list of undefined colors in a file. And then, specify the undefined color file, which you saved just before, in the "Color table file" option to "Convert..." again.

(If the "Display undefined color" is left specified, the "Undefined color" dialog appears again. Press "Ignore".) Colors are allocated according to the color table file in a newly created RSD.

LABP: Converts Labp-formatted file to binary data

Version 1.12

Documentation date: 3/2/98

Description

Converts a labp-formatted file to binary data.

Running the C preprocessor in conjunction with the labp command permits a more intuitive way of describing HMD data.

How to use

```
labp [options] labp-file >out-file
labp [options] <labp-file >out-file
```

The labp command takes a single file as input and writes the result on stdout.

An important design objective of the labp command was to reduce the complexity of handling pointers when HMD data is created.

The labp command can be used as an assembler for generating HMD data.

The output of the xhmd command (xhmd = HMD reverse assembler, see the xhmd document) can also be used as input for the labp command. This allows existing HMD binary data to be updated.

The input file contains a symbolic description of the data. Specific values are automatically calculated so data can be easily added and deleted.

The following describes the syntax of the labp input file:

- **Label Definitions**
A character string that starts with a letter, is followed by a sequence of letters and digits and ends with a colon (:) is considered to be a label. The first label at the beginning of the input file is assigned address 0.
- **Numeric Values**
Numeric values can be represented as either hexadecimal or decimal integers. If a numeric value begins with a "0x" it is considered to be a hexadecimal number. Numeric values are expanded at the current address as 32-bit values, except for cases described below under Special Operators. When a numeric value is 32 bits, the address is incremented by 4.
- **Label References**
A character string that starts with a letter and is followed by a sequence of letters and digits is assumed to be a reference to a label. A reference to a label is replaced with the address that has been assigned to that label.
- **Operators**
'+', '-', '*', '/', '&', '|', and '~' can be used as operators.
There is no operator precedence. Parentheses should be used to indicate precedence.
- **Special Operators**
There are 4 special unary operators: 'B', 'H', 'W', and 'M'.
B(x) places the value of x into a byte (8 bits). The address is incremented by 1.
H(x) places the value of x into a halfword (16 bits). The address is incremented by 2.
W(x) places the value of x into a word (32 bits). The address is incremented by 4. This is the same behavior as if x alone were specified.

For $M(x)$, the MSB of the datum corresponding to the size of the expanded value of x is set to 1. For example, $M(B(x))$ expands into a byte with the 8th bit of x set to 1.

- Comments
Characters which follow `'/'` or `'#'` through the end of the line are ignored and treated as a comment.

A semicolon (`;`) must be placed at the end of each logical line.

The `hmd.def` file is provided to make it more convenient to describe HMD data.

Because the `hmd.def` file conforms to the format of the C preprocessor, you can use the following commands (when `hmd.def` is located in `/usr/local/include/hmd`).

```
/usr/lib/cpp -I /usr/local/include foo.lab llabp >foo.hmd      (IRIX)
```

```
%compiler%cpppsx -I /usr/local/include foo.lab llabp >foo.hmd  (PlayStation Programmer tools)
```

(please use the 32-bit version of `cpppsx`)

Running the C preprocessor in conjunction with the `labp` command permits a more intuitive way of describing HMD data.

In the example below, the input file describes a flat gradation triangle in HMD format.

```
#include <hmd/hmd.def>

HMD_VERSION;          // Version ID
0;                    // MAP FLAG
prim_hdr_sect / 4;
3;                    // num of prim blocks
0;                    // pre-process
prim1 / 4;
0;                    // post-process

coord_sect:
1;                    // num of coords
coord0:
0;                    // flg
COORDM(0,0,0         //rot
      0, 0, 0);      //trans
0;                    //super

prim_hdr_sect:
1;                    // num of hdrs

prim_hdr0:
4;                    // size of non-shared hdr
M(poly_sect / 4);
M(vert_sect / 4);
M(norm_sect / 4);
M(coord_sect / 4);

prim_sect:

prim1:
TERMINATE;           // next prim
prim_hdr0 / 4;
M(1);                // num of types

      DEV_ID(SCE) | CTG(CTG_POLY) | DRV(INI) | PRIM_TYPE(TRI | COL);
      H(2); M(H(1)); // size, num of data
      (poly0 - poly_sect) / 4; // polygon index

poly_sect:
```

```

poly0:
// Colored Flat No-Texture Triangle
B(128); B(0); B(0); B(0x30);      // R0, G0, B0, code
B(0); B(128); B(0); B(0x30);     // R1, G1, B1, code
B(0); B(0); B(128); B(0x30);     // R0, G0, B0, code
H(0); H(0);                       // norm0, vert0
H(1); H(2); // vert1, vert2

vert_sect:
SVECTOR(150, 150, 150); // vert0
SVECTOR(-150, -150, 150); // vert1
SVECTOR(150, -150, 150); // vert2

norm_sect:
SVECTOR(0, 0, -ONE); // norm0

```

In the example above, "HMD_VERSION", "TERMINATE", "DEV_ID", "MATRIX", "SVECTOR" and similar strings are C preprocessor macros defined in hmd.def that will be converted into a format suitable for the labp command.

Options

-v Print version number on stderr.

Notes

- If the syntax of the input file is invalid, labp will terminate abnormally. In this case, output is limited to what information is available.
- It is convenient to place the "hmd.def" file in the C preprocessor include path. Please see the C language user's manual for information on how to specify the include path.
- In some C language preprocessors, line feeds which are input during macro expansion can be deleted. Therefore, use of the comment symbols `"/**"` and `"*/"` which the C preprocessor deletes is recommended rather than the labp comment symbol `"//"`.
- The maximum number of labels which can be used in labp is 16,384.

Changes in Version 1.02

Error processing for handling the input of incorrect data has been added.

The number of labels which can be used has increased from the 1,024 in Version 1.00 to 16,384.

Changes in Version 1.10

The bug relating to the intensity of the unary operator and operator combination has been fixed.

The built-in RotMatrix and VectorNormalS operators are now supported.

The input file line number which is output along with the error message is now output before being passed through the C language preprocessor.

Changes in Version 1.12

The unary operator `'~'` was added for bit inversion. This operator is used from hmd.def 1.12 onwards.

MESHCONV: Creates MSH file for RSD extension

Version 1.2

Documentation Date: 3/2/98

Description

This software creates a MSH file which is an RSD extension file. By adding this file to the RSD file, it is possible to retain the polygon linkage information. Also, the MSH file generated is needed in order to create HMD data which includes linkage information.

How to use

meshconv [options] RSD-file

Creates an MSH file from an RSD format file. The default creates a filename identical with the input filename except the extension which is replaced with ".msh".

Options

- o *output-file* Specify the output file name. The default creates a filename identical with the input filename in the current directory except the extension which is replaced with ".msh". This becomes effective when an MSH file has been specified within the RSD file.
- min *X* Selects the lowest linked number. The default is 2.
- max *X* Selects the highest linked number. The default is ∞ .
- help, -? Displays a summary of options which can be selected and a simple usage method.
- ns Prevents the linkage of shared polygons.

Remarks

- The related files to be used must exist in the same directory. If they exist in separate directories, specify the relative path or the absolute path.
- The MSH file to be output takes into account the information displayed within the RSD file and generates an MSH file. Since HMD cannot express linkage which extends over multiple matrices, make sure that the COD file is included when creating the MSH file. If the COD file is not specified, the object will be recognized as having only one matrix.
- When the output file name already exists, the contents are overwritten. Attention must be paid as no warning will be given.
- This program is compatible with the following RSD format versions. Be aware that no other RSD file versions can be used.

RSD File	version 3.0	
PLY File	version 3.0	
MAT File	version 3.0	
GRP File	version 3.0	
MSH File	version 1.0	(Output file)
PVT File	version 1.0	
COD File	version 1.0	
OGP File	version 1.0	
MOT File	version 1.0	

In meshconv (version1.0) the 5 RSD, PLY, MAT, COD and OGP files are read. Therefore, at the present time, the versions of files other than these five are not necessary. Since there is the possibility that they will be read in the future, the latest version should be used for all.

Changes in Version 1.0

First Release. (Distributed with PlayStation Runtime Library 4.0. Released on the BBS on July 2nd)

Changes in Version 1.1

The bug which caused the format of the internal function return values to be incorrect has been fixed. (The bug caused occasional crashes and entry into endless loops)

Changes in Version 1.2

The comments which were output before the ID are no longer output.

MIMEFILT: Creates MIMe difference file from TMD

Version 1.51

Documentation date: 4/1/96

Description

This software reads in multiple TMD files and creates the MIMe difference file.

How to use

MIMEFILT. [options] BASE.TMD VARIANT1.TMD VARIANT2.TMD...

Input of multiple TMD formats examines differences between the first TMD and the second and following TMDs to collect them in a difference file.

Differences between corresponding vertex coordinates and between normal vectors can be examined. The default creates only the vertex difference file (.VDF).

The wildcard is available for the argument. Omission of ".TMD" of the filename is allowed.

Options

- o *output-file* Specify an output filename. This filename is used for the vertex difference filename (.VDF) and normal difference filename (.NDF). The default gives a filename identical with the original TMD filename except the extension which is removed.
- n Creates a normal difference file (.NDF). The default is OFF.
- v Outputs detailed information about conversion to a standard output. The default is OFF.

Main improvements from previous version (ver. 1.5)

It is now possible to use the wildcard in the argument (with DOS).

Supplement

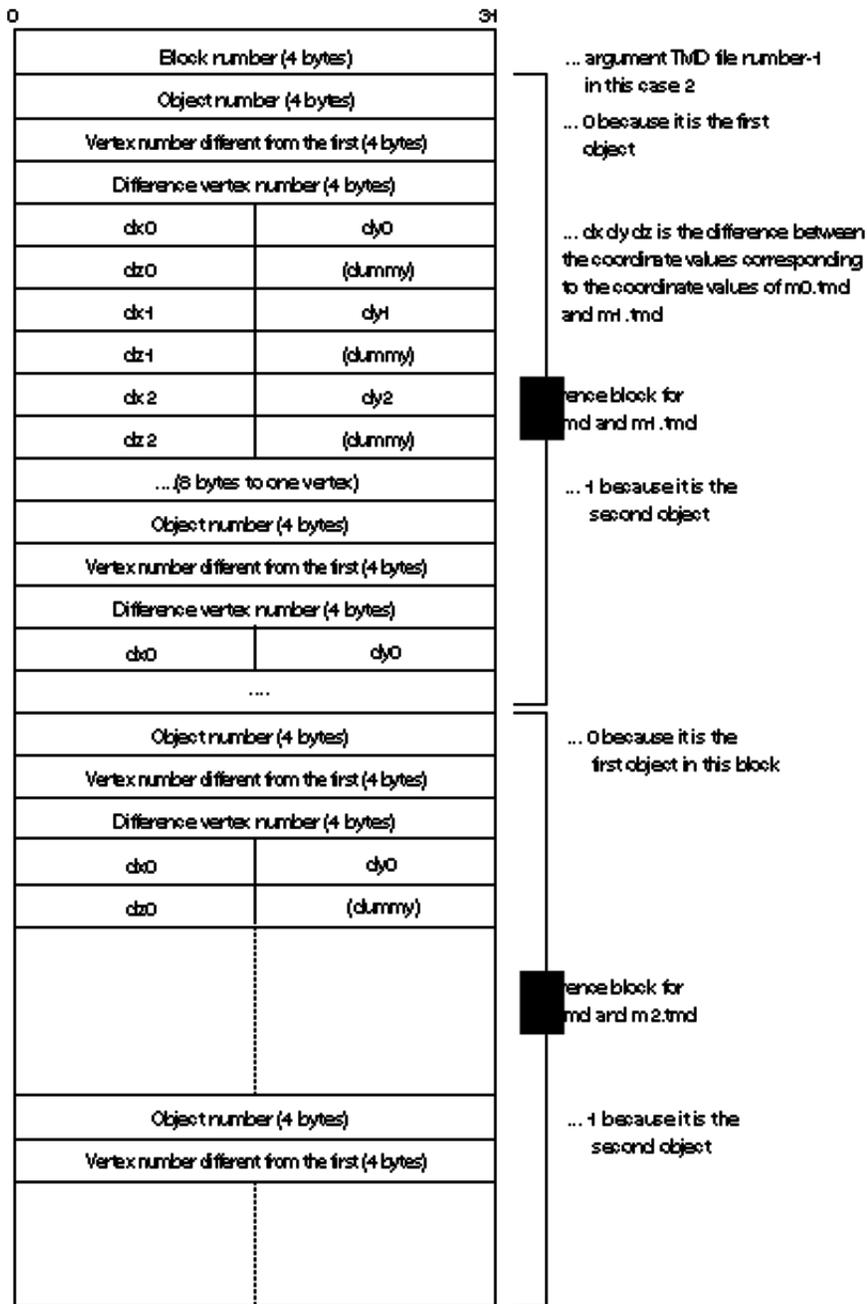
Numbers of vertices of all argument TMD files must be equal to each other. To create each TMD file, take care not to change the number and order of vertices. Create a basic prototype model with a 3D modeler which consists of sufficiently detailed polygons then move planes and vertices of the model to create a deformation model. (The number and order of vertices are not changed.) Save each model in the DXF format and convert it into the RSD format with DXF2RSD and then apply RSDLINK to it separately to create the TMD file.

File format

Format of the difference file created by MIMEFILT is the binary format in which blocks having the header and difference information for each object are coupled. (The normal difference file has the same format.)

```
C:> MIMEFILT MO.TMD M1.TMD M2.TMD
```

Figure 2-7: VDF File Format



MIMESORT: Sorts TMD Vertex Information

Version 2.1

Documentation date: 6/13/96

Description

This software reads in multiple TMD files and sorts vertex information. Use of this software before execution of MIMEFILT reduces the difference file. This lets the MIMe animation speed be higher.

How to use

MIMESORT [options] BASE.TMD VARIANT1.TMD[VARIANT2.TMD [VARIANT3.TMD...]]

This sorts vertex information in the TMD file and creates a new TMD file in order to reduce a MIMe difference file to be created by MIMEFILT. This allows reduction of calculation time for the MIMe animation run time.

As MIMESORT changes the order of the vertex information, all files to be filtered by MIMEFILT must also be sorted by MIMESORT. Sorting partly only some files with MIMESORT does not produce a correct difference file because vertices between TMD files cannot be corresponded to each other.

The wildcard is available for the argument.

When the -m option is not used, mimesort automatically sorts the vertices according to the vertex information within the model file. If the number of variation model files (TMD files other than base-model) exceeds 14, however, the specified models from the 15th onward are not used for sorting. As only the first 14 models are used as information for sorting, vertex information is rearranged on the models from the 15th onward. In other words, the variation model files from the 15th onward may possibly be made inefficient during output. Use of the -m option is recommended when the number of variation models is large.

To obtain the best possible results, please refer to the remarks section.

Options

- d *directory* Specify an output directory in which you want to place the sorted TMD file. The default is `“..STMD”`. You must create an output directory beforehand.
- v Outputs the order of vertices before and after sorting to an standard output. This allows you to check how the sorting was. The default outputs nothing. Meanings of outputs are as follows:
 - file: (vertex: a/b)
 - file: filename
 - a: Number of vertices contained in the difference file
 - b: Total number of vertices
 - .: Vertices which have no difference and are not contained in the difference file
 - +: Vertices which have no difference but are contained in the difference file
 - *: Vertices which have differences and are contained in the difference file

Supplement

Does not sort any other data than the vertex data (for example, normal data).

Does not always reduce the difference file. In order to make sure of MIMESORT's effect, filter data both before and after MIMESORT with MIMEFILT to check the size of the difference file. The smaller the size is, the more efficient the data is.

An input TMD file has to be a file which can be filtered with MIMEFILT. MIMEDORT does not deal with the TMD file which contains multiple objects.

The wildcard is available for the filename. However, specify it so that the base file comes in the head of the file. If a file which has to be a variant file is found in the base file position, an expected sorting result is not acquired. Check whether the wildcard works as it is expected or not with a message "base files=" which is given at execution.

Creates files just below the output directory even if a sub-directory is specified in the base or variant filename. Therefore, never specify an input file which has the same filename in a different directory.

Never specify the current directory such as "." to the output directory. If it is specified, an output file overwrites and damages the input file.

Be aware of the restriction in the number of characters in the MS-DOS command line.
Data other than vertex data (such as normal data, etc.) cannot be sorted.

Notes

When handling multiple deformation model files:

The use of MIMe for key frame animations where key frames for models are arranged in time series may often result in an enormous number of MIMe parameters. The concurrent application of too many parameters to a model consumes much time and memory, however. The replacement of the base model for key frame animations at places where only a few key frames are correlated with one another leads to efficiency enhancement.

If there are still more deformation files, use the -m option. Also use the -m option if you are not satisfied with the automatic vertex sort results.

Changes in version 1.1

The bug regarding the handling of a TMD file with a large PRIMITIVE section has been corrected.

Vertices are rearranged for 15 to 32 variation model files

Changes in version 1.2

The -stdin option enables the standard input of a model file name.

Changes in version 1.3

The bug whereby a primitive other than a triangle polygon could not be handled properly has been corrected.

Changes in version 2.0

A mode (-m option) which rearranges the vertices according to the sort table designated by the user has been added. Over 15 deformation model files can be handled when using this -m option,

Changes in version 2.1

The message output bug has been fixed and the output message has been standardized with the English.

MKMIME: Makes MIMe for HMD

Version 1.12

Documentation date: 10/26/98

Description

Reads multiple HMD files and creates an HMD file which includes MIMe information.

How to use

`mkmime [options] hmd0 hmd1 [hmd2...] >out-file`

The `mkmime` command takes 2 or more HMD files, finds the differences between them and writes an HMD file to stdout that contains the MIMe information. One file is used as the base MIMe and additional files are use as the MIMe key. `mkmime` is similar to the `mimfilt` command which obtains difference information for TMD files. However, `mkmime` not only writes out differences but outputs completely formatted HMD data as well. Differences, vertices, normals, and coordinates can all be calculated.

If no options are set, the differences of vertices and coordinates are obtained and HMD data is created as Vertex-MIMe, Joint-Axes-MIMe.

Options

- | | |
|--------------------|--|
| <code>-v</code> | Print version number on stderr. |
| <code>+vert</code> | Generate Vertex-MIMe information.
When vertex information is the same in all specified HMD files, information pertaining to Vertex-MIMe is not generated even if this option is specified.
This is the default setting. |
| <code>-vert</code> | Inhibit the generation of Vertex-MIMe information.
Vertex differences are ignored within the specified HMD file. |
| <code>+norm</code> | Generate Normal-MIMe information.
When normal information is the same in all specified HMD files, information pertaining to Normal-MIMe is not generated even if this option is specified. |
| <code>-norm</code> | Inhibit Normal-MIMe information.
Normal differences are ignored within the specified HMD file.
This is the default setting. |
| <code>+axes</code> | Generate Joint-Axes-MIMe information.
When coordinate information is the same in all specified HMD files, information pertaining to Axes MIMe is not generated even if this option is specified.
When this option is specified, the <code>-rpy</code> option discussed below automatically becomes effective and the output of Joint-RPY-MIMe information is suppressed.
This is the default setting. |
| <code>-axes</code> | Inhibit the generation of Joint-Axes-MIMe information.
Coordinate differences are ignored within the specified HMD file. |
| <code>+rpy</code> | Generate Joint-RPY-MIMe information.
When coordinate information is the same in all specified HMD files, information pertaining to Joint-RPY-MIMe is not generated even if this option is specified.
When this option is specified, the <code>-axes</code> option discussed above automatically becomes effective and the output of Joint-Axes-MIMe information is suppressed. |
| <code>-rpy</code> | Inhibit the generation of Joint-RPY-MIMe information.
Coordinate differences are ignored within the specified HMD file. |

This is the default setting.

Notes

- If the contents of the input HMD files are invalid, mkmime will terminate abnormally. In this case, output is limited to what information is available.
- All input HMD files must be the same size.
- The MIMePr section is not created. Within the output HMD, the data word which contains the MIMEID of the MIMe primitive header and the MIMEPR PTR are set to 0. In this case, the MIMe parameter must be provided by the application program.
- MIMe information is added at the end of the HMD input file.
- When it is necessary to place data other than MIMe (e.g. image data) at the end of HMD data, and the xhmd command is used to disassemble the output HMD file, the HMD file must be restructured using labp after performing appropriate post editing. (xhmd and labp are described in separate documents).

Changes in Version 1.02

- A bug which made Shared Polygon Primitives (other than those lined to post-process) unable to process, was fixed.

Changes in Version 1.10

- This software is now compatible with the HMD MIMe format which was changed from PlayStation library version 4.1 onwards. Please note that the HMD file created by mkmime 1.10 is not compatible with libraries prior to library 4.0.

Changes in Version 1.11

- A bug that sometimes caused abnormal termination when mkmime was activated with no arguments specified, was fixed.

Changes in Version 1.12

- An bug that occasionally caused an abnormal termination in *assertion failed* was fixed.

MKTOD: Makes TOD from PRJ and HRC files

Version 1.4

Documentation date: 5/1/96

Description

This software reads in the project information (PRJ file) and hierarchy information (HRC file) to create the layout file (TOD file).

How to use

MKTOD [options] hrc-file...

MKTOD is executed in the project directory and reads in the project file (MODEL.PRJ and OBJECT.RPJ) and the HRC file which is given with the argument to create the TOD file in the TOD directory. The TOD file has information about the layout of multiple objects. Unless the TOD directory exists in the current directory, a TOD file is created in the current directory. In both case, both project files must exist in the current file.

The HRC file which is the argument must exist in the current directory (.\) or .\HRC unless a path is specified explicitly. MKTOD searches the HRC file in this order. Omission of the extension ".HRC" is allowed.

Options

- o output-file* Specify an output TOD filename. The default creates a filename identical with the argument HRC filename in the current directory except the extension which is replaced with ".TOD".
- tree* Outputs the hierarchical structure of object to a standard output.
- v* Outputs information about what is being converted to a standard output.

Changes from Version 1.3

Previously there were cases where part of the origin was not output properly, but this has now been corrected.

An SGI version has been added.

Constraints

The current version has the following constraints:

Skips sequence blocks in the HRC file. Therefore, the TOD file which is created by MKTOD is always for a frame (that is, it is not an animation data but layout data.)

Note

Never use the parallel move (-t) or scale-up/down (-s) option of RSDLINK to create a program which uses the TOD file outputted by an animation tool or MKTOD. Specifying the move or scale-up/down option in conversion from the RSD to the TMD causes wrong movement or size of object because the movement and scale factor for the RSD object are stored in the TOD file. Execute the move or scale-up/down with the RSD data with RSDFORM beforehand.

Examples

Example 1: Display by the -v option

MKTOD is usually started up in a directory (project directory) having sub-directories shown below. The animation tool creates this directory structure and the project file.

```
C:> CD \HOME\PROJ1
C:> DIR A^W
      [.]    [..]    [HRC]  [ORG]  [RSD]  [CAM]
      [LHT]  [ACT]  [TMD]  [TIM]  [TOD]
      OBJECT.PRJ  MODEL.PRJ
C:> MKTOD -v TEST.HRC
      Input file   : C:\HOME\PROJ1\HRC\test.hrc
      Output file  : C:\HOME\PROJ1\TOD\test.tod
      Reading ...  C:\HOME\PROJ1\model.prj
      Reading ...  C:\HOME\PROJ1\object.prj
      Reading ...  C:\HOME\PROJ1\HRC\test.hrc
      Number of Frames : 1
      Number of Objects : 21
      Number of Packets : 72
      Output file size : 1096 bytes
```

As mentioned above, the -v option lets you know output and input filenames, numbers of frames, objects and packets and file size of the created TOD file.

Example 2: Display by the -tree option

The following is a display of the parentage between objects with the -tree option. Names and IDs of all objects written in the hierarchical structure description part of the HRC file are displayed. The child object is indented from the parent object in it.

```
C:> MKTOD -tree HAND.HRC
      = world(#999)
        +- joint1(#1001)
          +- obj1(#101)
          +- joint2(#1002)
            +- obj2(#102)
        +- joint3(#1003)
          +- obj3(#103)
          +- joint4(#1004)
            +- obj4(#104)
        +- joint5(#1005)
          +- obj5(#105)
          +- joint6(#1006)
            +- obj6(#106)
        +- joint7(#1007)
          +- obj7(#107)
          +- joint8(#1008)
            +- obj8(#108)
        +- joint9(#1009)
          +- obj9(#109)
          +- joint10(#1010)
            +- obj10(#110)
```

The above is an example when you input the following HRC file (ID is available from the OBJECT.PRJ file.)

```
C:> TYPE HRC\HAND.HRC
@HRC940401
```

Number of objects

21

Instance description part

```

world      null
obj1       cubel.rsd
obj2       cubel.rsd
obj3       cubel.rsd
obj4       cubel.rsd
obj5       cubel.rsd
obj6       cubel.rsd
obj7       cubel.rsd
obj8       cubel.rsd
obj9       cubel.rsd
obj10      cubel.rsd
joint1     null
joint2     null
joint3     null
joint4     null
joint5     null
joint6     null
joint7     null
joint8     null
joint9     null
joint10    null

```

Matrix (rotation angle, scale factor, parallel move) initialization part

```

world 0
      0 0 0 1.0 1.0 1.0 0 0 8000
obj1  0
      0 0 0 4.5 4.0 1.0 0 -200 0

<...intermediate parts are omitted...>

joint9 0
      0 0 0 1.0 1.0 1.0 -500 -950 0
joint10 0
      0 0 0 1.0 1.0 1.0 0 -650 0

```

ACT file description part (sequence block)

```
0
```

Hierarchical structure description part

```

world      joint1
joint1     joint2 joint3 joint5 joint7 joint9
joint3     joint4
joint5     joint6
joint7     joint8
joint9     joint10
joint1     obj1
joint2     obj2
joint3     obj3
joint4     obj4
joint5     obj5
joint6     obj6
joint7     obj7
joint8     obj8
joint9     obj9
joint10    obj10

```

RSD2DXF: Converts RSD to DXF

Version 1.10

Documentation date: 9/1/95

Description

This software converts the RSD format file group into the DXF format. This is an MS-DOS Commandline application.

How to use

RSD2DXF[options] [ARG-files...]RSD-files...

A filename of the DXF to be outputted is identical with the input RSD filename except the extension which is replaced with ".DXF". The wildcard is available for the argument.

Options

- o *output-file* Specify an output filename. The default creates a filename identical with the input filename in the current directory except the extension which is replaced with ".DXF". Use of this option is allowed only when only one RSD file is inputted.
- r Reverses the front and rear of all polygons that are contained in a DXF file to be created.
- h Shows a list of options which can be specified and an easy-to-use method.
- v Shows information about conversion.
- ARG-files Specify an argument file which describes options for RSD2DXF.EXE, input filename and others. With this, the filename having the extension ".ARG" is constructed as the argument file. Use this when the command argument of above 128 bytes is required.
The line starting with "#" is regarded as comment and has no effect on execution of RSD2DXF.EXE.

Remarks

In a coordinate system of the DXF file to be created, the right side is +X, the upper side is +Y and the rear side is +Z. It is identical with the default which DXF2RSD.EXE and DXFRSDW.EXE suppose as the coordinate system of the DXF file. Therefore, when you read in and edit the DXF file to be created with a modeler, in order to let the model be subject to the coordinate system of the modeler, rotation of the model is required.

Information about the normal, color and texture contained in the RSD file is discarded.

If the same output filename exists, the contents are overwritten. No warning will be given so pay attention to it.

Even if a argument filename is written in the argument file, it is meaningless.

The wildcard is unavailable in the argument file.

RSD2HMD: Converts RSD to HMD

Version 1.5

Documentation date: 3/2/98

Description

rsd2hmd converts an RSD format file into HMD format.

How to use

rsd2hmd [options] RSD-file

Converts an RSD format file to HMD format. ".hmd" is added to the input file name from which the extension has been removed to create the file name of the HMD file which will be output.

Options

<i>-o output file</i>	Specify an output file name. The default creates a filename identical with the input filename in the current directory, except that the extension is changed to "hmd".
<i>-s X</i>	Enlarges or reduces the object using the specified scaling factor.
<i>-help,-?</i>	Displays a list of options which can be specified and an easy-to-use method.
<i>-bin2cl file-name</i>	Outputs the output HMD data expressed as Hexadecimal notations to the specified file.
<i>-ni</i>	Does not output image data to HMD.

Remarks

- The RSD file and TIM file used must exist in the same directory. If they exist in separate directories, specify the relative path or the absolute path.
- The output HMD file contains all the information displayed within the RSD file. If information which you do not wish to include such as linkage information or COORDINATE information is present, either do not perform file specification within the RSD file or comment it out.
- When the same output file name already exists, the contents will be overwritten. Attention must be paid as no warning will be given.
- This program is compatible with the following RSD format versions. Be aware that no other RSD file versions can be used.

RSD File	version 3.0
PLY File	version 3.0
MAT File	version 3.0
GRP File	version 3.0
	(as the GRP file is not used in rsd2hmd version 1.5, the version is not needed)
MSH File	version 1.0
PVT File	version 1.0
COD File	version 1.0
OGP File	version 1.0
MOT File	version 1.0

Changes in Version 1.0

First release. (Distributed with PlayStation Runtime Library 4.0)

Changes in Version 1.1

- The bug which caused the format of the internal function return values to be incorrect has been fixed.
- The bug which output a meaningless address from the CLUT TOP section of the primitive global header when there was no TIM file CLUT has been fixed.
- The bug which caused the color of a textured polygon with no light source calculation to be output incorrectly has been fixed.

Changes in Version 1.2

- The bug in which the word MSB which displays the number of sequences and size of the Sequence Pointer within the Animation Primitive was not set, has been fixed.

Changes in Version 1.3

- The bug which caused the Block section Primitive Top to be incorrectly output when selecting the -ni option, has been fixed.

Changes in Version 1.4

- RSD files containing Colored Flat Texture Quads, Colored Gouraud Texture Quads and Preset Gouraud Texture Quads can now be converted correctly into HMD.
- PLY files not containing normals can now be read.
- Fixed a bug which caused invalid data to be generated when certain material types were specified for conversion from RSD to HMD and light-source calculation was not specified.
- Fixed a bug which caused all animation type INI bits to be set when more than one MOT file was loaded.
- When light-source calculation is not specified, using F for MAT shading will produce Gouraud output for material types G, H, N. This is consistent with the actions performed by rsdlink.
- When RSD data with material type T, W is converted to HMD, and light-source calculation is not specified, output is flat texture (R:128, G:128, B:128) with no light-source calculation.

Changes in Version 1.5

- The "END" instruction, indicating the end of an animation sequence, can now be converted correctly into HMD.

RSDCAT: Links multiple RSD files to create new RSD file set

Version 1.06

Documentation Date: 8/18/97

Description

This software links multiple RSD file sets to create a new RSD file set. This is an MS-DOS Commandline application.

How to use

RSDCAT [options] [ARG-files...] RSD-files...

The wildcard is available for the argument.

Options

- o *output-file* Specify an output filename. The default creates "OUT.RSD", "OUT.PLY", "OUT.MAT" and "OUT.GRP" in the current directory.
Specifying a filename with this option creates outputs identical with the input filename except the extension which is replaced with RSD, PLY, MAT and GRP.
- h Shows a list of options which can be specified and an easy-to-use method.
- v Shows information about conversion.
- ARG-files Specify an argument file which describes options for RSDCAT.EXE, input filename and others. With this, the filename having the extension ".ARG" is constructed as the argument file. Use this when the command argument of above 128 bytes is required.
The line starting with "#" is regarded as comment and has no effect on execution of RSDCAT.EXE.

Remarks

A group is added to a group file (*.GRP) to be created for each input RSD file. This group consists of all polygons of the input RSD file and has a group name identical with the input RSD filename except the extension which is removed.

RSDCAT.EXE does not compress overlapping textures, vertexes, normals, groups and others through multiple input RSD files. Therefore, files which are outputted by RSDCAT.EXE may require some amendment.

If the same output filename exists, the contents are overwritten. No warning will be given. Pay attention to it.

Even if an argument filename is written in the argument file, it is meaningless.

The wildcard is unavailable in the argument file.

Changes in Version 1.01

- The bug in which the output GRP file was incorrect has been corrected.

Changes in Version 1.02

- The command argument can now be specified in the argument file.

Changes in Version 1.03

- The version of the outputted MAT file has been changed to @MAT940801

Changes in Version 1.04

- Now compatible with Material Type D and H (gradation polygon).

Changes in Version 1.05

- Now compatible with file names containing two or more periods.

Changes in Version 1.06

- The bug in which processing was not correctly carried out when a flag other than a decimal was described in the MAT file has been fixed.

RSDFORM: Deforms or moves RSD data

Version 1.81

Documentation date: 4/1/96

Description

This software deforms or moves the 3D model data (RSD) for the artist.

How to use

RSDFORM [options] RSD-name

This scales up/down, rotates and moves a RSD format 3D model data to deform the shape of the model. When multiple options are specified at a time, they will be executed in the order of scaling → rotation → parallel move. Omission of the extension “.RSD” of the argument RSD filename is allowed.

Options

- o *RSD-name* Specify an output RSD filename. The extension is removed. No overwriting to the input file is allowed. The default is “a”.
- s *x y z* Specify scale factors for x, y and z direction. Specify them with the floating-point numeric value. The value of 1.0 is a scale of 1/1. Specifying a negative number is allowed. In this case, a mirror image (shape at the symmetric position) with respect to the axis is acquired.
- r *x y z* Specify rotation angles for the x, y and z axes. Specify them with the floating-point numeric value. The plus value represents the right handed screw direction against the axis (*). The default’s angle unit is the degree but once the -rad option is specified, it turns to the radian.
 (*) “Right handed screw direction” means a direction to which when you erect the thumb of the right hand and direct its tip to the plus direction of the rotation axis, the remaining fingers wind.
- t *x y z* Specify parallel moves for the x, y and z axes. Specify them with the floating numeric value.
- rad Turns the unit of the rotation angle to the radian.
- l Fixes the center of the model for scaling and rotation. Specify this together with the -s and/or -r option.
- c Shares a file, if possible. (Refer to Supplement.)
- c[*pmg*] Specify any of PLY, MAT and GRP files to be shared. Combine three characters (p, m and g) to specify them like -cpg (this is equal to -cp -cg). When other options are specified and a file cannot be shared (refer to Supplement), a warning is given and the sharing option is ignored.
- k Holds the original file version if the input RSD has an old version ID. Unless this option is specified, @RSD940102, @PLY940102, @MAT940801 and @GRP940102 are given.
- quiet Adds no comment of deformation history to the output file. The default is OFF.
- v Outputs detailed information about conversion to an standard output.

Improvements from previous version (ver. 1.8)

When the RSD in another drive’s current directory is designated it can now be correctly read (with DOS).

When a file described within an RSD file cannot be located, first convert the file name to all capital letters and then search. If the file still cannot be located, convert the file name to all lower-case letters and search (only with SGI).

Supplement

The RSD data is a set of four files, .RSD, .PLY, .MAT and .GRP. These files must exist in the same directory.

Each file can be shared when the following condition is met:

- PLY file: No deformation is specified.
- MAT file: No mirror image against the original data
- GRP file: Can always be shared.

Provided that all files to be shared must exist in the same directory.

When you create a program which uses the TOD file, use this tool but RSDLINK for moving or scaling up/down. (Refer to Chapter 2, Section 9: RSDLINK.EXE)

The PlayStation coordinate system uses the sign 16-bit. This rule does not apply to the RSD but you must convert it into the TMD with RSKLINK so that it can settle in the coordinate system.

Examples

Example 1: Basic use

Scale up it in the x axis direction by 10 times and then rotate it around the x axis by 45 degrees. Specify the file sharing if possible.

```
C:> RSDFORM -v -s 10 1 1 -r 45 0 0 -c CUBE
```

Output	Explanation
Input RSD : CUBE	
Output RSD : a	
Scale : 10, 1, 1	
Rotation : 45, 0, 0 (degree)	change specification definition
Translation : 0, 0, 0	
Range x : +0.0 ... +1000.0 (center: +500.0)	size before
y : -500.0 ... +500.0 (+0.0)	change
z : -500.0 ... +500.0 (+0.0)	(coordinate values based on maximum and minimum axis values)
CUBE.ply → a.ply	I/O file name
CUBE.mat → CUBE.mat (Shared)	makes MAT file
CUBE.grp → CUBE.grp (Shared)	GRP file sharable
CUBE.rsd → a.rsd	
New Range x : +0.0 ...+50000.0 (center: +500.0)	size after change
y : -707.1 ... +707.1 (+0.0)	(coordinate values
z : -707.1 ... +707.1 (+0.0)	based on maximum and minimum axis values)

Example 2: To deform locally

Apply the -l option in the example 1 deformation. With this option, you can specify the center of gravity of the model as the center of rotation and scaling. Location of the model does not change. Compare this with the result of the example 1.

```

C:> RSDFORM -v -l -s 10 1 1 -r 45 0 0 CUBE
Input   RSD   :   CUBE
Output  RSD   :   a
Scale   :    10,    1,    1
Rotation :    45,    0,    0 (degree)
Translation :    0,    0,    0
Range   x   :   +0.0 ... +1000.0 (center: +500.0) \
          y   :   -500.0 ... +500.0 (           +0.0)  —
          z   :   -500.0 ... +500.0 (           +0.0) / \
                                           \ \
                                           \ \
CUBE.ply →      a.ply
CUBE.mat →      a.mat
CUBE.grp →      a.grp
CUBE.rsd →      a.rsd
                                           / \
                                           / \
New Range x   :  -4500.0 ... +5500.0 ( center: +500.0) \ /
          y   :   -707.1 ... +707.1 (           +0.0)  —
          z   :   -707.1 ... +707.1 (           +0.0) /

```

Example 3: Copying the RSD file

This software can be used as a copying tool for the RSD file. This is also available for change of the RSD name. It is useful for you to create the following batch file.

```

C:> TYPE RSDCOPY.BAT
      @ECHO OFF
      RSDFORM -o %2 %1

```

Example 4: Overwriting of the RSD file

This tool protects the pre-deformation RSD file from being overwritten. However, you can use this tool as an overwriting tool with the following batch file.

```

@ECHO OFF
SET ARGS=
REM
REM          reads all arguments
:LOOP1
IF "%9"==" " GOTO LABEL1
SET ARGS=%ARGS% %1
SHIFT
GOTO LOOP1
:LABEL1
REM
REM          immediately changes name to _tmp
REM
rsdform -o _tmp %ARGS% %1 %2 %3 %4 %5 %6 %7 %8
IF ERRORLEVEL 1 GOTO END
REM
REM          searches for last argument (=I/O file name)
REM
:LOOP2
IF "%1"==" " GOTO LABEL2
SHIFT
GOTO LOOP2
:LABEL2
REM
REM          overwrites input file name (%0)
REM
rsdform -o %0 -quiet _tmp
REM
REM _tmp.*      completes deletion of starred files (*)
REM
DEL _tmp.*
:END
SET ARGS=

```

RSDLINK: Converts RSD to TMD

Version 3.72

Documentation date: 10/1/96

Description

This software converts the 3D model data (RSD) for the artist into the 3D model format (TMD) which the libgs can deal with.

How to use

RSDLINK [options] rsd-names...

RSDLINK [options] rsd-names [options] rsd-names [options]...

RSDLINK [options] arg-files...

RSDLINK [options] prj-file

These link multiple RSD data you specify with the argument to create a TMD file. This allows you to set a scale factor and parallel move for each RSD data. (Refer to example 1.)

Unless an argument RSD name contains a path, first the current directory (.) and then the .\RSD directory are searched.

If there are many arguments, you can write them in a file (*.ARG) to give them. Omission of ".RSD" is allowed but ".ARG" of the argument filename must not be omitted. (Refer to example 2.)

Similarly, you can give directly a project file (*.PRJ). This case is equal to a case when you give all of the RSDs contained in the project file with the argument. At the same time, the -id option is regarded to be specified. Omission of ".PRJ" is not allowed.

Options

- o *file-name* Specify an output filename. The default is "a.tmd".
- s *factor* Scales up or down the RSD data after the next argument (*).
- sc *factor* The scale factor is rounded to powers of 2. The default is 1.0. Set a proper scale factor so that it can settle in the TMD format coordinate system because the coordinate values in the TMD format are 16-bit integers.
- t *x y x* Moves the RSD data after the next argument (*). The default is (0 0 0).
- id *PRJ-file* Specifying the project file (MODEL.PRJ) creates an array of object ID numbers for each object to be created in the TMD file. The result is created in the current directory as a header file to the C language. The output filename is identical with the TMD filename except the extension which is replaced with ".h" (refer to example 3.)
- info Outputs very detailed information about the kind, vertex coordinate values and textures of objects which are being converted to an standard output (refer to example 5.)
- v Outputs detailed information about conversion such as the number of polygons to a standard output. Moreover, as their approximate sizes (Range: minimum and maximum values of the vertices (x, y and z)) in the PlayStation coordinate system are also outputted, you can check their positions or sizes before displaying them on the PlayStation board (refer to example 4.)

(* **Note:** The -s and -t are effective to all the RSDs which appear after them unless they are specified again.

Main improvements from previous version (ver. 3.65)

Now able to link more RSD. Although the maximum up until now was around 500 RSD, it is now possible to link as many as the memory will allow. With DOS, although it depends on the available conventional memory space, it is possible to link up to 2000 RSD.

When the file described in the RSD file cannot be found, convert the file name to all caps and search again. If the file still cannot be found, convert the name to all lower case and search again (only for SGI versions).

Constraints

The current version has the following constraints:

May not link the RSD data if there are too many polygons in a RSD data. The upper limit of the number of polygons depends on the numbers of vertices and normals and the free memory capacity but it is usually about 5000 polygons. However, this limit applies only to "a RSD" and there is, in practice, no limit to the number of the RSDs to be linked and number of polygons of the whole TMD to be created.

If a RSD which does not appear in the project file is linked, "-1" is allocated to the header file.

If you specify the same RSD twice when you specify a project file, the ID value is allocated to only the first RSD and "-1" is allocated to the second and following RSDs.

A line of the RSD file must consist of 1024 bytes or less.

Note

Never use the parallel move (-t) or scale-up/down (-s) option to create a program which uses the TOD file outputted by an animation tool or MKTOD. Specifying the move or scale-up/down option in conversion from the RSD to the TMD causes wrong movement or size of object because the movement and scale factor for the RSD object are stored in the TOD file.

For the move and scale-up/down, use DXF2RSD (-sc -t) or RSDFORM (-s -t) in the RSD format, unless otherwise specified. This allows more accurate and easy-to-use data to be obtained.

rsdlink performs file input/output in small units. As a result, it can take a long period of time when designating a network file as the input/output file. When linking a large RSD data file, please use the local disk as much as possible.

Supplement

The image data file (TIM file) which is described in the RSD file must exist in the same directory (.\) as the RSD file exists or ..\TIM directory. RSDLINK searches the TIM file in this order.

The following table shows what TMD type the RSD is converted to by RSDLINK. Heads and items show types which are described in the MAT file. (Tri, Quad, Line, Sprite and others are specified with numerals in the PLY file.) The 0x... is a header of the TMD packet. In the table, only one-sided polygons which are not semitransparent are shown. Packet of the semitransparent or both sided polygons is the same though the header has a different value. For details refer to "Appendix: File format RSD and RMD".

Table 2-3: RSD → TMD Conversion Comparison Table

		With light source calculation		Without light source calculation	
Shading →		Flat	Gouraud	Flat	Gouraud
Material + polygon type	C+Tri	0x20000304	0x30000405	0x21010304	(0x21010304)
	G+Tri	0x20040505	0x30040505	(0x31010505)	0x31010505
	T+Tri	0x24000507	0x34000509	(0x25010507)	(0x25010507)
	D+Tri	(0x24000507)	(0x34000509)	0x25010507	(0x25010507)
	H+Tri	(0x24000507)	(0x34000509)	(0x35010509)	0x35010509
	C+Quad	0x28000405	0x38000505	0x29010305	(0x29010305)
	G+Quad	0x28040705	0x38040505	(0x39010505)	0x39010505
	T+Quad	0x2c000709	0x3c00080c	(0x2d010709)	(0x2d010709)
	D+Quad	(0x2c000709)	(0x3c00080c)	0x2d010709	(0x2d010709)
	H+Quad	(0x2c000709)	(0x3c00080c)	(0x3d01080c)	0x3d01080c
	C+Line	(0x40010203)	(0x40010203)	0x40010203	(0x40010203)
	G+Line	(0x50010304)	(0x50010304)	(0x50010304)	0x50010304
	T+Sprite	(0x64010305)	(0x64010305)	0x64010305	(0x64010305)
1x1	(0x6c010204)	(0x6c010204)	0x6c010204	(0x6c010204)	
5x5	(0x74010204)	(0x74010204)	0x74010204	(0x74010204)	
15x15	(0x7c010204)	(0x7c010204)	0x7c010204	(0x7c010204)	

TMDs mentioned in the parentheses are for improper combinations of material types. Some options are ignored to change them to proper TMDs.

Examples

Example 1: Use

```
C:> RSDLINK -v -o BOXES.TMD BOX1 -s 2.0 -t 100 100 100
      BOX1 BOX2 -t 200 -200 200 BOX3
```

In this case, four objects shown below are put together into a TMD file (BOXES.TMD).

BOX1: BOX1 is an equimultiple

BOX1': BOX1 is doubled (100 100 100) and moved

BOX2: BOX2 is doubled (100 100 100) and moved

BOX3: BOX3 is doubled (200 200 200) and moved

Example 2: How to put together arguments in a file

You can also put together arguments in a file as follows.

```
C:> TYPE TEST.ARG
      BOX1
      -s 2.0 -t 100 100 100 BOX1 BOX2
      -t 200 -200 200 BOX3
C:> RSDLINK -v -o BOXES.TMD TEST.ARG
```

Example 3: Read-in of project file and creation of ID array

Project file ("MODEL.PRJ") is an ID table of objects which an animation tool creates.

```
C:> TYPE MODEL.PRJ
@PRJ940701
# Next available ID
  6
# IDObject Name
  0   head.rsd
  1   body.rsd
  2   l_arm.rsd
  3   r_arm.rsd
  4   l_leg.rsd
  5   r_leg.rsd
```

If there is a file mentioned above and you execute

```
C:> RSDLINK -o ROBO.TMD MODEL.PRJ
```

or

```
C:> RSDLINK -o ROBO.TMD -id MODEL.PRJ BODY HEAD L_ARM R_ARM
L_LEG R_LEG
```

ROBO.H having the following contents is created together with ROBO.TMD. IDs of objects are registered as elements of the array in the same order they appear in the TMD. You can use this when you want to access each object in the TMD from the application program.

```
C:> TYPE ROBO.H
/*
 * TMD ID List for "ROBO.TMD"
 *
 * Created by rsdlink(v3.6) at 10:10 09/16/94
 *
 * (C)1994 Sony Computer Entertainment Inc. All Rights Reserved.
 */
int robo_list[] = {
    1, /* body */
    0, /* head */
    2, /* l_arm */
    3, /* r_arm */
    4, /* l_leg */
    5, /* r_leg */
};
```


Example 5: Outputs when the -info option is used

You can check what TMD the data is actually converted to with the -info option.

```
C:> RSDLINK -info BOX
```

Output	Explanation
Input RSDs : 1 object(s)	number of objects in TMD
RSD[0] = "BOX"	RSD name
Total vertices : 8	total number of vertices
Total normals : 12	total number of normal lines
Total primitives : 12	total number of primitives
-----<box>-----	
= [0] FLAT TEX 3-POLY (0x24000507) LIGHT: ON =	
Vert-0: (-150, -150, -150) (#2)	
Vert-1: (-150, -150, -150) (#6)	
Vert-2: (-150, -150, -150) (#0)	
Norm-0: (0, 0, -4096) (#0)	
UV 0-2: (0 0) (47 0) (0 47)	
Pixel mode : 4bit CLUT : (x,y)=(0 480)	
Texture Page: 10 Texture No. : 0	
= [1] FLAT TEX 3-POLY (0x24000507) LIGHT: ON=	
.	
.	
.	

For each primitive,

```
= [0] FLAT TEX 3-POLY(0x24000507) LIGHT: ON =
```

that is, the [number of polygons], FLAT/GOURAUD, TEX/NO TEX, SEMITRANSARENCY ON/OFF, TWO-SIDED/ONE-SIDED, GRADATION, types of primitive (3-POLY, 4-POLY, SPRITE), hexadecimal display of primitive header (0x...) and LIGHT calculation ON/OFF are displayed.

Then, coordinate values of each vertex which forms the primitive are displayed as shown below.

```
Vert-0: (-150, -150, -150) (#2)
```

(#...) is a vertex number used in the PLY file.

Similarly, the normal is displayed.

```
Norm-0: (0, 0, -4096) (#0)
```

The RSD normal is usually normalized to size 1 (in this case, (0, 0, -1.0)) but as the floating point in the TMD has a 4096 value for 1 of the RSD, it is displayed as mentioned above.

Then, material information such as UV coordinates and color of the texture is displayed.

RSDUP: Converts old RSD file to HMD-compatible RSD file

Version 1.1

Documentation date: 3/2/98

Description

Converts an old RSD-formatted file to a new RSD-formatted file which conforms to HMD.

How to use

rsdup [options] RSD files ...
rsdup [options] -bat TXT file

An old RSD-formatted file is converted to a new RSD-formatted file which conforms to HMD. rsdup reads multiple RSD files containing components of a single character and collects them together in a single RSD file. The default output files are a.rsd, a.ply, a.mat and a.cod. Described in the COD file is the component of the RSD file which contains the COORDINATE of the unit matrix. After the COD file is output, the initial location can be determined by editing the file.

Options

<i>-o output-file</i>	Specifies the base name of the output file. If an extension is included it is ignored. The default base name of the output file is "a".
<i>-ogp</i>	Outputs an OGP file. The default output file name is a.ogp. If the <i>-o</i> option is used, a different base file name can be specified. The object group name takes the extension of each RSD file that is read.
<i>-pvt PVT-file</i>	Specifies the PVT file which is included in the RSD file.
<i>-cod COD-file</i>	Specifies the COD file which is included in the RSD file. In this case, the <i>-ogp</i> option is invalid and the output file "a.cod" is not written as the default.
<i>-mot number-of-MOT-files MOT-file...</i>	Specifies the MOT files which are included in the RSD file. A single MOT file must contain the same animation type.

Notes

- RSD files and TIM files must be in the same directory. If they are not, the relative path or absolute path must be specified.
- If the output file already exists, the contents are overwritten. No alert is displayed so please exercise care.
- When the *-bat* option is specified, the TXT file, which is the last parameter to rsdup, is read. The target RSD files which will be read are described in the TXT file. In this case, one file must be specified per line. (a TXT file can be output efficiently using `ls *.rsd > debug.txt`)
- As with the old RSD format, check to be sure there is a 1:1 correspondence for every index when the COD file and MOT file are specified. If a 1:1 correspondence does not exist, the RSD file will not be converted properly to HMD.

- The versions of RSD format that can be used with this program are shown below. Please check the version numbers beforehand as other versions of RSD cannot be used.

RSD File version 2.1

PLY File version 2.1

MAT File version 2.1, version 2.2

GRP File version 2.1

(In rsdup version 1.0, the GRP file is not used so the version is not checked.)

Changes in version 1.0

- Initial release.

Changes in version 1.1

- Comments before ID are no longer output.
- PLY files that do not contain Normals can now be loaded.
- When multiple (identical) RSDs specify the same TIM, the output RSD will use a single TIM instead of using Includes for separate TIMs.

TMD2PMD: Converts TMD to PMD

Version 1.14

Documentation date: 10/16/95

Description

Input of the TMD format file (*.TMD) which is the PlayStation 3D modeling data format creates the PMD file (*.PMD) which is the high speed modeling data format. For detail of PMD format, refer to "Appendix: File format: PMD". This software is an MS-DOS Commandline application.

How to use

TMD2PMD [options] TMD-files...

The argument can contain wildcard identifiers. Multiple files can be given to perform batch conversions. In such cases, each file is converted to the same file name with a .pmd extension and is created in the current directory.

Options

<i>-o output-file</i>	Specifies the output filename. The default creates a filename identical with the input filename in the current directory, except the extension which is replaced with ".PMD". Input of two or more files does not allow use of this option. If you specify the wildcard which applies to two or more files, use of this option causes an error.
<i>-h</i>	Shows a list of options which can be specified and an easy-to-use method.
<i>-v</i>	Shows detailed information about conversion.
<i>-s</i>	Sets vertex data in the PMD file to be created in sharing vertex mode.
<i>+s</i>	Sets vertex data in the PMD file to be created in independent vertex mode. (Default)
<i>-l</i>	Turns ON the light calculation flag in the PMD file to be created.
<i>+l</i>	Turns OFF the light calculation flag in the PMD file to be created. (Default)
<i>-b</i>	Turns ON the Back clip flag in the PMD file to be created. (Default)
<i>+b</i>	Turns OFF the Back clip flag in the PMD file to be created.
<i>-notge</i>	Turns OFF the TGE flag of all packets in the PMD file to be created.
<i>+notge</i>	For packet TGE flag in the PMD file to be created, those in the input TMD file are copied. (Default)

Remarks

In the TMD file, texture polygons with light source calculation do not include color information. tmd2pmd sets this sort of data as a brightness of 0x80 together with R, G, B when it creates the PMD file.

Changes in Version 1.10

- Fixed a bug in which the ABE and TGE bits in the TMD packet were not reflected in the output PMD.

Changes in Version 1.11

- Added an option (-notge) so that the TGE bit in a TMD packet is ignored and the TGE bit in the output PMD is forcibly set to 0.

Changes in Version 1.13

- Changed so that the directory which creates working files can be specified by the TMP environmental variable. Fixed a bug in which some types of polygons did not have correct vertex colors in the created PMD file.

Changes in Version 1.14

- Fixed a bug in which TMD files containing only gradation polygons were not converted correctly.

TMSORT/INFO: Sorts and displays TMD files

Version 1.25

Documentation date: 4/1/96

Description

tmsort - This software sorts the 3D model data file (TMD) by packet type. This minimizes branching by packet type at painting. This is used to realize a high painting speed.

tmdinfo - Displays the contents of the TMD file.

How to use

TMSORT [options] TMD-file

TMDINFO [-v | -l] TMD-file

This sorts packets in each object in the TMD file you specified with the argument and outputs the result into a new TMD file. Also, the total number of each packet type is output to a standard output. Omission of ".TMD" is allowed.

Options

-o file-name Specify an output TMD filename. The default is "a.tmd". However, overwriting to the input file is not allowed.

-n Executes no sorting but outputs information about the input TMD to a standard output.

-v or -l Outputs very detailed information about the type, vertex coordinate values and texture of all packets to a standard output. The display correspond to the output of RSDLINK-info. Refer to rsdlink.txt.

If this tool is copied and the name is changed to tmdinfo.exe (tmdinfo with SGI) it will be made into a state in which the *-n* option has been specified in the default. Also, it will be possible to output the total number of vertices and normals along with their location, size, etc. in the PS coordinate system. Although no sorting is carried out, this is useful in confirming TMD file contents. Only the *-v* (or *-l*) option can be specified.

Main improvements over previous version (ver. 1.1)

The location and size ("Range: minimum/maximum value of vertex (x,y,z)") in the PS coordinate system of the input TMD can now be output (tmdinfo only).

Constraints

The number of vertices (and normals) per object must be 8189 or less when the *-v* option is used. Unless the *-v* option is used, this constraint is meaningless.

Sorting is executed, object by object in the TMD. A different object is not sorted even if it exists in the same TMD.

XHMD: Converts an HMD data file to labp format

Version 1.20

Documentation date: 10/26/98

Description

Converts an HMD data file to labp format.

The following description applies to Version 1.20.

How to use

```
xhmd [options] labp-file >out-file
```

The xhmd command takes a single file as input and writes the result on stdout.

When used with the labp command (labp = label processor, HMD assembler.

Please see the labp document), HMD data can be updated relatively simply.

General workflow is shown below using IRIX as an example.

```
$ xhmd foo.hmd >foo.lab
$ vi foo.lab                                # Edit HMD data
$ /usr/lib/cpp -I /usr/local/include foo.lab >foo2.hmd # Create updated HMD data (example using UNIX).
                                                    # /usr/local/include/hmd/hmd.def exists.
```

Options

`-v` Print version information on stderr.

Notes

- If the HMD input file contains invalid pointer information, the xhmd command will terminate abnormally. In this case, output is limited to what information is available.
- Stdin cannot be used as input.
- Polygon section analysis is not performed. Information that is output is an enumeration of simple data.

Changes in Version 1.02

- The bug which made it impossible for any Shared Polygon Primitives, except for those linked to post-process to be processed, has been fixed.
- In order to make it compatible with the partial C language pre-processor, the format of the output comments has been changed from labp format to C language pre-processor. Please refer to the labp documentation for details.
- Error processing when incorrect data is input has been added.

Changes in Version 1.10

- Now compatible with the HMD MIME Format and Ground Primitives which were changed in PlayStation Library 4.1.

Changes in Version 1.11

- The mistake in the output result which incorrectly spelt "SPLINE" as "SPRINE" was corrected.

Changes in Version 1.12

- The upper eight bits of the polygon index in the polygon and shared polygon primitives can now be interpreted as a polygon division parameter.

Changes in Version 1.15

- Now compatible with the animation formats based on the six types of rotational order introduced in Library 4.2.5.

Changes in Version 1.16

- It is possible to specify the size of an image primitive header with no CLUT as 1, however a bug which caused this to be recognized as an error was fixed.

Changes in Version 1.20

- Reverse assembling of polygon data is now possible.
- Processing when multiple labels are assigned to the same address is now possible.

Chapter 3:

2D Graphics Conversion Utilities

BMP2TIM: Converts BMP to TIM

Version 2.2

Documentation date: 1/25/96

Description

This software converts the Windows BMP file into the PlayStation image data file (TIM).

How to use

BMP2TIM [options] BMP-files ...

Input of the Windows 3.0 BMP format image data file creates the PlayStation TIM file. The argument can contain wildcard identifiers. Multiple files can be given to perform batch conversions. Omission of the ".BMP".

Options

- o output-file* Specify an output filename. The default is to use the input filename with the extension '.tim' and write it to the current directory.
- org x y* Specify the VRAM image data area origin. The default is (0,0).
- plt x y* Specify the VRAM pallet (CLUT) area position. The default is (0,480).
- t* Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default is for all colors to be non-transparent.
- b* Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black.
- v* Shows detailed information about conversion.

Major changes from previous version 2.1

This version allows creation of a TIM with no CLUT from a Windows BMP file. To do this, use the "-plt x y" option with a negative x or y: for example:

```
bmp2tim -plt -1 -1 a.bmp
```

Major changes from versions prior to 2.1

Now correctly converts even when the width is not 4x.

Even when an absolute path is provided for an input image file (if not explicitly specified with the -o option), a TIM file is created in the current directory.

The -mode option was eliminated. The file is converted to the mode closest to the size of the input BMP data pixels. Please use Photoshop or something similar to reduce the number of colors.

Changed so that the -t option can be set as semi-transparent for (0 0 0). In the previous version the default for (0 0 0) was transparent, but in this version it is displayed as black. If sprite areas other than characters are to be made transparent, (with Photoshop or something similar), specify the -b option for those areas which are to remain black when converting.

Constraints

The BMP compressed format is unavailable for the current version.

PICT2TIM: Converts PICT to TIM

Version 3.1

Documentation date: 10/17/94

Description

This software converts the Macintosh PICT file into the PlayStation image data file (TIM).

How to use

PICT2TIM [options] PICT-files ...

Input of the Macintosh PICT format image data file creates the PlayStation TIM file. The wildcard is available for the argument. You can input and convert multiple files at a time.

Options

- | | |
|-----------------------|---|
| <i>-o output-file</i> | Specify an output filename. The default creates a filename identical with the input filename at the current directory, except extension which is converted into the ".TIM". |
| <i>-org x y</i> | Specify the VRAM image data area origin. The default is (0,0). |
| <i>-plt x y</i> | Specify the VRAM pallet (CLUT) area position. The default is (0,480). |
| <i>-t</i> | Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default lets all colors turn nontransparent. |
| <i>-b</i> | Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black. |
| <i>-v</i> | Shows detailed information about conversion. |

Constraints

The current version has the following constraints:

- Convertible PICT data is limited to the Pixmap
- 32-bit PICT data cannot be converted
- JPEG compressed PICT data cannot be converted

RGB2TIM: Converts RGB to TIM

Version 2.1

Documentation date: 4/1/96

Description

This software converts the RGB format image data file into the PlayStation image data file (TIM).

How to use

RGB2TIM-size w h [options] input-file

Input of the RGB format image data file creates the PlayStation TIM file. For -size, you must specify numbers of pixels for the width and height of the input image. The input data is the 24-bit interleaved RGB data. The wildcard is available for the argument but you can convert only one file at a time. Although multiple RGB data can be converted together, since options such as size, bit mode, etc. become common to all data, individual output file names cannot be specified.

Options

-o <i>output-file</i>	Specify an output filename. The default creates a filename identical with the input filename at the current directory, except extension which is converted into the ".TIM".
-org <i>x y</i>	Specify the VRAM image data area origin. The default is (0,0).
-plt <i>x y</i>	Specify the VRAM pallet (CLUT) area position. The default is (0,480).
-mode <i>4 8 16 24</i>	Specify a pixel size for the output TIM data. The default is the 16-bit mode.
-t	Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default lets all colors turn nontransparent.
-b	Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black.
-skip <i>n</i>	Specify the number of bytes which are skipped from the head of file. The default is 0.
-v	Shows detailed information about conversion.

Main changes from previous version

Creation of a TIM file with no CLUT is now possible (-plt option).

The -size option correctly specifies width only and when the height is given a value less than 0 , the height is automatically calculated. Cases where only the height is correctly specified are treated similarly.

Multiple RGB data can now be converted together. However, common options are applied.

Supplement

Too many colors of the RGB data which is inputted may cause conversion of it to 4 or 8-bit mode to be impossible. In such case, reduce beforehand the number of colors with Photoshop and others.

TIMTOOL

Working with projects

Each project contains a list of all the TIM images currently loaded, as well as information such as the Graphics Mode currently used. The project information changes when you add or remove a TIM image from the project. If the project information has changed, when you select to start a new project, save the current project or exit the application, you will be asked if you want to save the current project.

Note: If any of the TIM images have been modified, you will firstly be asked if you wish to save the modified TIM images. Modifications include changes made to the TIM or CLUT position, and changes made to the SemiTransparency of any of the CLUT elements.

Loading TIM images into the current project

In order to load TIM images into the current project, you can select either the 'Add Tim(s)' button, the 'Tim Options', 'Add to List' menu item or the 'Ctrl + A' shortcut. The standard Windows '95 'Open File' dialog box will appear. From here, you can select a single TIM image file by clicking on it, or multiple TIM image files by either using the shift and ctrl keys, or by dragging a selection area around all the files you wish to load.

Note: Any directories selected will be ignored.

Positioning TIM images within the Frame Buffer

The easiest way to position TIM images within the frame buffer is to hold the left mouse button down over the TIM image within the Frame Buffer, and drag the image to the required position before releasing the mouse button. Notice that the image will snap to both the Texture Page sides and to all sides of existing TIM images. This makes positioning of the TIM images very easy.

Another way of positioning the TIM images is to select the TIM image (either by clicking on it in the frame buffer or selecting its name from the tree view) and modifying the X and Y text box entries for the TIM image.

Note: There is no automatic sort of TIM images within the frame buffer. A generic sort procedure would not work for all developers as there are usually a number of conditions that must be taken into account for the sort, (i.e., the three small grass textures and the road textures must be located within the same texture page, the three title images must be located at the same position etc.). Rather than having an interface whereby the user could enter this information into the program before sorting, and maintain it for additions and deletions to the project's image list, the interface was enhanced to enable the user to quickly be able to correctly place the images where they pleased (i.e., drag and drop, snapping etc.).

Positioning the CLUTs of 4-bit and 8-bit TIM images within the Frame Buffer

One way of positioning the CLUTs of 4-bit or 8-bit TIM images is to select the TIM image (either by clicking on it in the frame buffer or selecting its name from the tree view) and modifying the CLUT X and Y text box entries for the TIM image.

Autosort CLUTs

If you have a lot of CLUTs to position, you can arrange them all into a selectable area. In order to do this, select an area in the frame buffer by holding down the Shift key and dragging an area with the mouse in the frame buffer or texture page preview area. The selection will be snapped to a suitable scale. Select the 'Clut Options', 'Autosort 4-bit CLUTs into selected area' in order to automatically sort the 4-bit CLUTs, or select the 'Clut Options', 'Autosort 8-bit CLUTs into selected area' in order to automatically sort the 8-bit CLUTs.

If you have not selected enough area in order to be able to sort all of the CLUT's, an error message will be displayed and the sorting will be cancelled.

Note: Any 4-bit TIM images with CLUTs greater than 16 x 1, or any 8-bit TIM images with CLUTs greater than 256 x 1 will be ignored during this sort.

Note: The automatic sort for 8-bit CLUTs will ignore the width of the selection if it is under 256 pixels in width.

Importing and converting Windows BMP, JPEG, TIFF and PCX files

TIM Tool can import images from the following image file formats:

- Windows BMP Image
- JPEG Image
- TIFF Image
- PCX Image

In order to import and convert an image of one of the above types, select the 'File', 'Import Image' menu item. A dialog box will appear, allowing you to select one or more image files.

Note: If you select an image that is not aligned properly (e.g., 8-bit images need to be of an even width), the converter will automatically align the converted TIM image properly.

If you select a single image, the first of the two image areas to the right of the screen will display a scaled version of the image. The second image will display the image after it has been converted to the selected image depth (either 4, 8, 16 or 24-bit). The conversion will check to see if a TIM image file exists with the same root name (e.g., if converting forest.bmp, the program will check for a file called forest.tim). If the file exists, the conversion extracts the image and CLUT position and the previously converted color depth from the existing TIM image file.

If you select multiple images, the first of the two image areas to the right of the screen will display a scaled version of the last image to be selected. The second image will display this image after it has been converted to the selected image depth (either 4, 8, 16 or 24-bit).

If you select the 'Set for Black' option from the 'Semi Transparent Information' area, all of the colors in the image with the value 0 will have their STP bit set.

If you select the 'Set for all Colors except Black' option from the 'Semi Transparent Information' area, all of the colors in the image with a value not equal to 0 will have their STP bit set.

If you select both the 'Set for Black' option and the 'Set for all Colors except Black' from the 'Semi Transparent Information' area, all of the colors in the image will have their STP bit set.

If you select a single image to convert to a 16-bit TIM image file, the 'Use Alpha Mask Image for Transparency' check box becomes selectable. This allows you to select an Alpha Mask Image file which will be used to set the STP bits within previously selected TIM image file. The Alpha Mask Image file should preferably contain black and white information, although the conversion uses the average of the RGB value of each pixel to decide whether the pixel in the Alpha Mask Image file should be interpreted as black or white. Black areas will set the STP bit, whereas white areas will leave the STP bit clear. The Alpha Mask Image file must be the same size as the image file to be imported.

Semi Transparent Information

Color information for the PlayStation is stored in 1555 format (1 bit for the SemiTransparent information (STP) and 5 bits each for the RGB information). The color 0 is ignored by the PlayStation hardware (it is used for 'full' transparency). In order to be able to use the color black as a drawn color, you can set the STP bit of all colors with the value 0, so that the hardware doesn't see it as the value for 'full' transparency any more.

However, if you are already using the STP bit for semi transparency (e.g., the program has called the SetSemiTrans() function), this value will be picked up as a semi-transparent black. In order to get a non-transparent 'pure' black in this case is to leave the STP bit clear and modify the Red and/or Green and/or Blue information so that the color is nearly black (e.g., 1 for Red, 0 for Green and 0 for Blue). The human eye will not notice the difference, but fortunately the PlayStation hardware will.

Modifying the STP bits of 4-bit and 8-bit TIM images

If you have loaded a 4-bit or 8-bit TIM file, you can view the images CLUT and modify the STP of each CLUT entry. Select the required 4-bit or 8-bit TIM image from the frame buffer or tree view and either right-click the TIM Preview Image area and choose the 'CLUT Options' menu item, or select the 'CLUT Options', 'CLUT Options' menu item.

The 'CLUT Options' dialog box will appear, showing the current CLUT for the selected image in the lower left hand corner, and a preview of the image in the lower right hand corner. From this dialog, you can set the STP of individual CLUT entries by either double-clicking the CLUT entry, or selecting the CLUT entry and clicking the 'Semi Transparent' check box. This will modify the TIM image, so when you want to save your changes, select the 'TIM Options', 'Save Modified TIM's' menu item.

The 'CLUT Position' area enables you to temporarily render the TIM image using a different CLUT, specified by modifying the X and Y position edit boxes.

Saving out the information to a parsable text file

If you select the 'TIM Options', 'Save Map File' menu option, a dialog box appears prompting you for a file name for the map file. This map file contains parsable text information about all the TIM image files currently loaded into the project. A parser can be created extremely easily, especially if the programmer has tools that work with Windows .INI files.

An example of the file format is as follows:

```
[TIM_1]
File=T:\TEXTURES\Barier2.tim
X=976
Y=64
Width=16
Height=16
ColourDepth=4
ClutX=864
ClutY=192

[TIM_2]
File=T:\TEXTURES\Chimney.tim
X=956
Y=128
Width=4
Height=64
ColourDepth=4
ClutX=880
ClutY=192

[TIM_3]
File=T:\TEXTURES\Clump.tim
X=1008
Y=64
Width=16
Height=64
ColourDepth=4
ClutX=864
ClutY=193
```

Saving all the images as one or more large TIM files to speed up loading times

Loading a lot of separate image files from the CD can be fairly slow. In order to speed up loading, an image of the frame buffer can be built up and saved as a large single image file. TIM Tool has a feature which allows you to perform this easily.

If you select an area in the frame buffer (by holding down the 'Ctrl' button and dragging an area with the mouse), and select the 'TIM Options', 'Save area as 16-bit TIM' menu option, a dialog will appear prompting you for a filename for the TIM image file.

Figuring out what the Texture Page Preview is showing

The texture page preview shows what the PlayStation hardware sees when in the current texture page at the current color depth. If, for example, you have selected a texture page to preview in 8-bit, all the 8-bit images in that texture page will be displayed in the texture page preview. All other non 8-bit images will be displayed as gridded areas, indicating that some image is located there, allowing the user to snap to these images.

Note: When you click on an image in the frame buffer, the texture page automatically renders at the selected images' color depth.

Note: 24-bit images cannot be used as textures by the PlayStation hardware, and are therefore not displayed in the texture page preview image.

TIM2BMP: Converts TIM to BMP

Version 1.2

Documentation date: 6/22/98

Description

This software converts a PlayStation texture file into Windows image data.

How to use

tim2bmp [option] *input-file*...

Input of the PlayStation TIM file (*.TIM) creates a Windows 3.0 BMP format image data file.

Options

-c If this option is selected when the input data is in black and white binary 16bit mode, the 8 bit index format bit map file is output.

Limitations

Not compatible with BMP compressed format.

16bit mode input data is converted to 24bit map file.

The MS-DOS Wild card can be used when specifying the input-file.

Changes in Ver 1.1

- A bug which caused 4-bit mode and 8-bit mode TIM data to be converted incorrectly, was fixed.

Changes in Ver 1.2

- A bug which caused the extension not to be attached to the output file when converting data from a file having eight letters (not including the file name extension), was fixed.

TIMPOS: Changes VRAM in TIM

Version 1.1

Documentation date: 5/1/96

Description

This software changes the VRAM image and pallet (CLUT) addresses stored in the PlayStation image data file (TIM).

How to use

TIMPOS [options] input-file image-x image-y [clut-x clut-y]

For TIM file given by the argument, set the VRAM addresses to which the image and pallet (CLUT) are located. Addresses which you set are coordinate values of the top of the left side of the image and pallet. The VRAM space is 1024 wide x 512 long. Use of the argument does not allow you to set only the pallet VRAM address. Set the image VRAM address at the same time. If you enter a negative number for any address, the address is never changed. This allows you to change only the pallet VRAM address (Example 1). Also, if a negative number is given as a CLUT address, a TIM file with no CLUT will be created. When the CLUT address is not changed, specify only the image address (Example 2).

Omission of the extension ".TIM" of the TIM filename given by the argument is allowed.

Options

- o *output-file* Specify an output filename. The default is a filename identical with the input filename.
- v Outputs information about the input file and conversion to a standard output. Input of only the TIM file together with this option allows you to know the current address information.

Examples

Example 1: Changing only pallet address

```
C:>TIMPOS TEST.TIM -1-1 0500
```

Example 2:

```
timpos test.tim 640 0           (change only image address)
timpos test.tim 640 0 -1 0      (Create TIM with no CLUT)
```

Changes from previous version (ver. 1.0)

When a negative number is set as the CLUT address, a TIM file with no CLUT is created.

An SGI version has been added.

Note

A TIM file with no CLUT cannot be converted to a TIM file with a CLUT.

TIMUTIL: Allows multiple TIM, BMP, PICT and RGB conversions

Version 1.36

Documentation date: 4/3/95

Description

TIM utility is a Windows application which allows mutual conversion of the bit map formats of the PlayStation TIM, Windows BMP, Macintosh PICT and general-purpose RGB. Execute it by double clicking on the icon registered in either the file manager or program manager. Since timutil help can be displayed in the help menu, please refer to this for details.

Options

None

Notes

timutil uses SPIN.VBX. Install SPIN.VBX in the /WINDOWS/SYSTEM.

PICT which do not contain any mixed-type TIM, compressed format BMP, JPEG compressed PICT, or 32bit PICT bit maps cannot be read.

Mixed-type TIM, compressed format BMP, JPEG compressed PICT format cannot be written. Also, the bit depth of the write file is limited to 4,8,16 or 24 bits.

For example, with conversions which reduce the bit depth such as from 16bit data to 8 bit, the color information is approximated. At such times, since it approximates the color map allocated on average to each R, G and B, rather than to the processes such as color compression, the color resolution may be decreased.

As a general principle, files which have been read cannot be overwritten. However, since the read format and write formats are both TIM, overwriting is possible when only the image starting point or the palette starting point are changed. This function is equivalent to the command line version of timpos.exe.

Changes in Ver 1.10

- The TIM file can now display to the artist board (DTL-H200). This is equivalent to the command line version timview.exe.
- The timpos.exe functions have been unified.
- A 'Save all files' selection has been added to the file menu.

Changes in Ver 1.15

- The I/O address which is used to display the TIM file to the artist board (DTL-H200) can now be set in aboard.exe.
- 'Settings' has been added to the file menu.
- The conversion parameter default value can be changed.
- When switching between the 'TIM display' and 'Screen resolution' the display resolution will now be relocated to the middle of the screen.

Changes in Ver 1.20

- A 'Writing palette' list has been added to the 'Parameter settings window' and the multi-CLUT TIM can now be operated.
- The palette contents are now faithfully reflected when converting image files with identical bit lengths.

Changes in Ver 1.30

- 'TIM layout' has been added to the window menu.
- The TIM file image and palette location can now be graphically edited.

Changes in Ver 1.35

- The height of the 'Display area' in the 'TIM layout' dialog which was 240, has been changed in consideration of the double buffer so that the area displayed on screen now has a height of 480 (two vertical 240 height areas).
- The bug in 'TIM layout' which caused TIM files to be displayed abnormally when the original point of the image did not fit within the texture page boundary has been corrected.
- It is now possible to open RSD files. At that point all the TIM files included in the RSD file will operate as if they have been opened.
- The current directory can now be changed when a file has been opened within drag and drop from the file manager or from a previously-opened file list. As a result, the directory containing the files which were last successfully opened becomes the current directory (previously the current directory was changed only in cases where the file dialog was used when the file was opened from the 'Open' menu or when the directory was changed with 'Save all files').
- The bug in which an overwrite check to an identical file was not performed when the write palette was changed has been corrected.

Changes in Ver 1.36

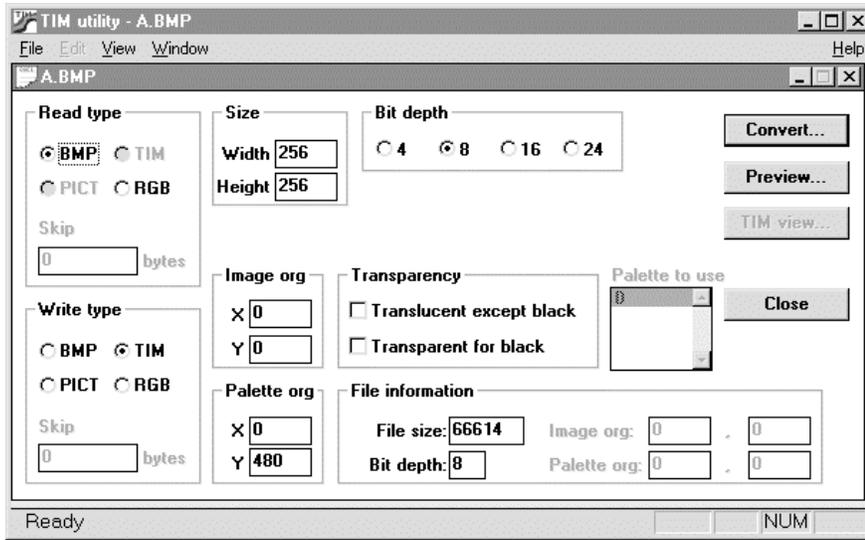
- The bug whereby a BMP file in which the number of colors used was lower than the number of colors calculated from the bit depth and could not be read has been corrected.

How to use

Selection of "Bit map file" sub-option from "Open..." option of the file menu lets you access the Parameter Setting window. Properly change parameter values and then press the "Convert..." button. Alternatively, you can select the "Give name to save..." option from the file name. The File Save dialog pops up. Entering a post-conversion filename starts format conversion.

You can drag an input file from the File Manager to drop it in the TIM utility window instead of using the "Open..." command.

Figure 3-1: Parameter Setting Window



Explanation of Parameter Setting window settings

- **Read-in format**
Shows the format of a file which was read in. When it shows one of the TIM, BMP and PICT format, you can select the RGB format. Header information of the TIM, BMP or PICT is ignored during conversion and the file is compulsorily read in as the RGB data.
- **"TIM"**
Is selected when a file which was read in can be constructed as the PlayStation TIM.
- **"BMP"**
Is selected when a file which was read in can be constructed as the Windows BMP.
- **"PICT"**
Is selected when a file which was read in can be constructed as the Macintosh PICT.
- **"RGB"**
Is selected when a file which was read in cannot be constructed as one of the TIM, BMP and PICT.
- **"SKIP"**
You can enter here when read-in format is the RGB. As many bytes as you specify here will be skipped when an input file will be read in as the RGB format.
- **Write-out format**
Select a post-conversion file format.
- **"TIM"**
Converts a file into the PlayStation TIM.
- **"BMP"**
Converts a file into the Windows BMP.
- **"PICT"**
Converts a file into the Macintosh PICT.
- **"RGB"**
Converts a file into the general-purpose RGB format. Selection of this button allows you to change values for the SKIP of the write-out format.
- **"SKIP"**
You can enter here when write-out format is the RGB. As many 0s as number of bytes you specify here will be added to when a file will be written out as the RGB format.

- **Size**
You can enter here when read-in format is the RGB. You must always enter here to construct byte string of the input file as the RGB format image data. If the image data size calculated from a value entered here and a value entered for the SKIP of the read-in format is larger than the size of a file which is inputted, it causes an error. If it is smaller, it lets a warning panel pop up.
- **Image origin**
You can enter here when write-out format is the TIM. Set the PlayStation VRAM image origin coordinates which are inserted in the output TIM file.
- **Pallet origin**
You can enter here when write-out format is the TIM and the bit depth is 8 or less. Set the PlayStation VRAM pallet (CLUT) origin coordinates which are inserted in the output TIM file.
- **Transparent control**
You can set the transparent control when write-out format is the TIM and the bit depth is 16 or less.

Selection of "Any other color than the black is semitransparent" option sets the transparent control bit to 1 for any pallet entry which has any other value than 0 for one of R, G and B.

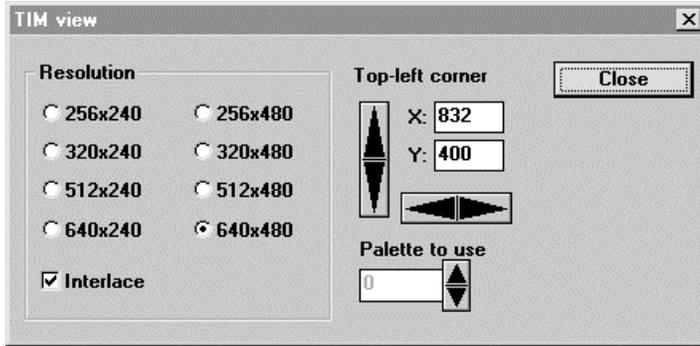
Selection of "Black is transparent" option sets the transparent control bit to 0 for any pallet entry which has 0s for all of R, G and B. Lack of this selection lets any color which has 0s for R, G and B turn nontransparent black.
- **Bit depth**
Specify the number of bits per pixel of the output file.

When write-out format is the BMP, you can specify one of 4, 8 and 24-bit. When it is the PICT, you can specify one of 4, 8 and 16-bit. When it is the RGB, only 24-bit is available.
- **Write-out pallet**
Select a pallet which is written out to a post-conversion file when read-in format is the TIM and multiple pallets are provided. Pallet which you select here will be also used in the Display and TIM display.

If you set "Reflect on origin" of "Write-out pallet" with "Set..." command of the file menu, Y coordinate value of the pallet origin increases or reduces automatically, depending on the place of the selected entry in the write-out pallet list.
- **Information about read-in file**
Shows the read-in file size and pixel depth.

It shows "Image origin" when read-in format is the TIM. Moreover, it also shows the "Pallet origin" when the "bit depth" is 8 or less.
- **Conversion**
Executes format conversion according to parameters you set. As the File Save dialog pops up, enter a post-conversion filename. This function is equal to "Give name to save..." option of the file menu.
- **Display**
Reads in a file according to the current read-in format and displays it. Colors are approximated when the bit depth of the read-in file is larger than that of a display you use. (The current version may require much processing time.)
- **TIM display**
Displays file contents on the Artist board when the current read-in format is the TIM. Pressing this button lets you access the following dialog.

Figure 3-2: TIM Display Dialog



This dialog allows you to set a screen resolution, a VRAM position corresponding to a video monitor screen connected to the Artist board and, in case of the TIM with multi palettes, a palette used for the Display, while the TIM file image being displayed on the Artist board.

When the bit depth is 4 or 8, the TIM display displays file contents by texture mapping to a polygon. Moving the display area allows you to view directly 4 or 8-bit index data or the pallet area.

When the bit depth is 4 or 8, the TIM file image data is transmitted to the image origin, as a rule, but when X of the origin is below 640, it is compulsorily transmitted to (640,y). This prevents it from overlapping with the display area. The pallet data (CLUT) is transmitted to the pallet origin. When the bit depth is 16 or 24, the TIM display displays file contents by transmitting a bit pattern to the image origin.

The display area is initialized so that the image is situated at the center of the 640 x 480 display screen.

- **Close**
Closes the current window.

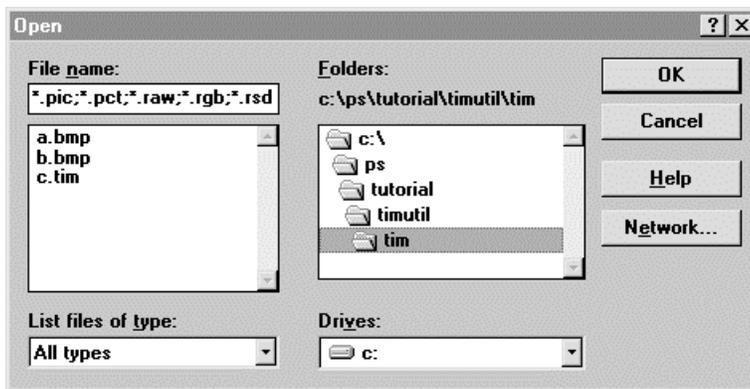
This function is equal to "Close" of the file menu.

Menu bar

"Open..." command ([File] menu)

Opens existing bit map files. The following dialog box appears.

Figure 3-3: File Open Dialog



This dialog allows you to set the following options.

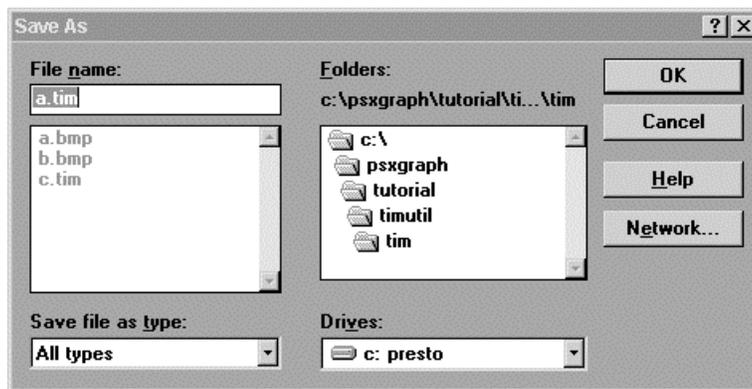
- **Filename**
Enter or select a bit map filename you want to open. List box shows files which have an extension corresponding to a file format you selected in the "File type" box.
- **File type**
Select a file format of bit map file you want to open.

TIM utility supports the PlayStation TIM, non-compressed format Windows BMP, PICT which includes one or more bit maps, except 32-bit mode and general-purpose RGB. (Write-out bit depth is limited to 4, 8, 16 or 24-bit.)

You can also select the RSD format model data. This opens all of the TIM data which are specified by the model data.

- **Drive**
Select a drive in which a bit map file you want to open will be saved.
- **Directory**
Select a directory in which a bit map file you want to open will be saved.
- **"Close" command ([File] menu)**
Closes a window which is currently active.
- **"Give name to save..." command ([File] menu)**
Give a filename to a bit map file you are dealing with to save it. As the following dialog box appears, you can give a filename to a bit map file you are dealing with to save it. This executes format conversion according to parameters you set.

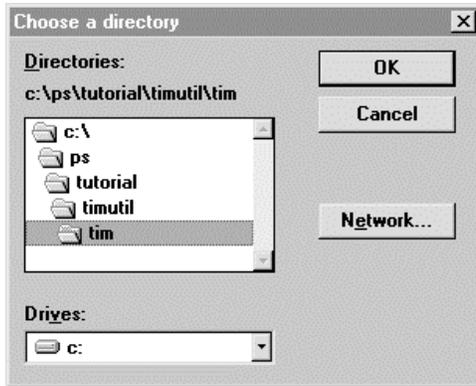
Figure 3-4: File Save Dialog



This dialog box allows you to set the following options to specify a filename or location in which you save it.

- **Filename**
As a rule, enter a name different from the original one. Give a filename of up to 8 half size characters and add an extension of 3 half size characters to it.
- **Drive**
Select a drive in which you save a bit map.
- **Directory**
Select a directory in which you save a bit map.
As a rule, overwriting to the read-in file is not allowed.
Only exception is the case in which both read-in format and write-out format are the TIM and no other setting than the image and pallet origins is changed. In this case, you can overwrite it. (This is equal to TIMPOS.EXE of the Comandline version.)
- **"Save all files..." command ([File] menu)**
Saves all bit maps that you are dealing with. As the following dialog box appears, specify a directory in which you write out them. This executes format conversion according to parameters set in each bit map. Filenames are identical with the original ones except extensions which are replaced by an extension specified in the write-out format.

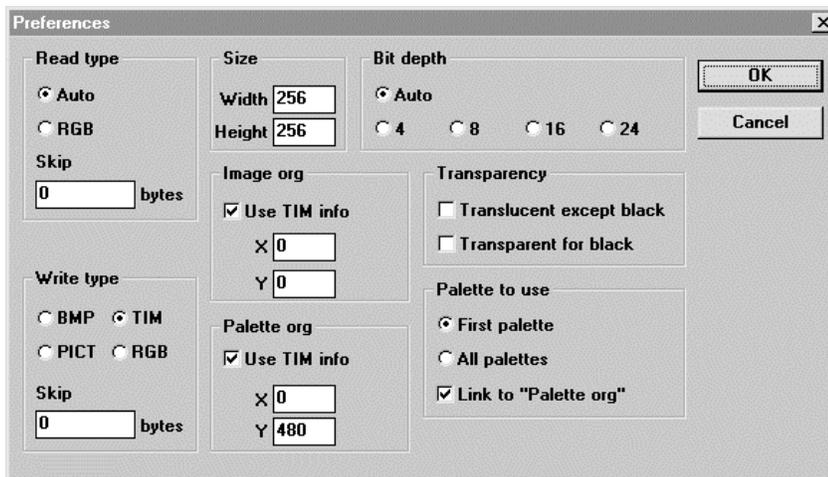
Figure 3-5: Directory Select Dialog



This dialog allows you to set the following options to specify locations in which you save bit maps.

- **Drive**
Select a drive in which you save bit maps.
- **Directory**
Select a directory in which you save bit maps.
As a rule, overwriting to the read-in files is not allowed.
If any write-out file conflicts with the rule, it causes an error and no conversion is executed at all.
- **"Set..." command ([File] menu)**
Sets parameter initial values which will be set in the Parameter Setting window when you will open a file.

Figure 3-6: Set Dialog



If "Subject to read-in file" is set in "Read-in format", a format which was judged from contents of the read-in file is set in the read-in format of the Parameter Setting window. If "RGB" is set, any file is set to the RGB. This also applies to the "Bit depth".

"Image origin" and "Pallet origin" are used as default values for conversion parameters when "TIM information priority" is not selected or when read-in format is any other format than the TIM.

For "Write-out pallet", you can select a write-out pallet from the write-out pallet list when a TIM file with multiple pallets (CLUT) is read in. Determine the default.

Selection of "First one" selects only No. 0 pallet. Selection of "Select all" selects all pallets as write-out pallet. The "Write-out pallet" is provided with another button, "Reflect on origin". This button sets whether the pallet origin is automatically set or not when an entry is selected from the write-out pallet list. If this is

set, selection of No. 0 entry from the write-out pallet list sets the pallet origin which is set in the original TIM file for the pallet origin. Selection of No. 1 entry adds 1 to Y coordinate of the pallet origin which is set in the original TIM file and sets the value for the pallet origin.

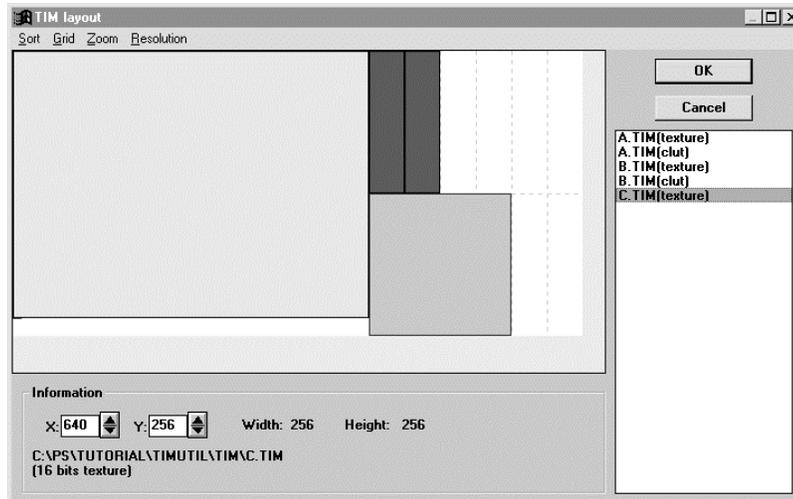
Other parameter values than mentioned above are simply used as the initial values in the Parameter Setting window.

Values set in the dialog will be saved at end of TIM utility.

"TIM Locate..." command ([Window] menu)

Opens "TIM Locate" dialog to locate graphically a TIM format file which is opened in the VRAM.

Figure 3-7: TIM Locate Dialog



"TIM Locate" dialog has, to divide roughly, 4 main areas shown below:

1. VRAM image area
2. Information area
3. Select list area
4. Menu bar

Figure 3-8: VRAM Image Area



This area displays the VRAM image on PlayStation. Rectangles which show image or pallet (CLUT) of the currently opened TIM file, a rectangle which shows a range which is displayed and texture page borderlines which are referred to for location are displayed. For image or pallet rectangle, the part which does not overlap with another image or pallet rectangle or the range rectangle is displayed in green and the part which overlaps with other rectangles or projects from the VRAM area is displayed in red. The left button of the mouse allows you to select these rectangles and you can move them by dragging. Use of the left

button of the mouse while pressing the Shift key or Ctrl key allows you to select multiple image or pallet rectangles. Rectangles you select are displayed in blue.

Figure 3-9: Information Area



This area displays information about selected image or pallet rectangle. Image or pallet origin coordinates, a VRAM range the rectangle occupies, a filename and the bit depth are displayed.

You can change the origin coordinates by clicking Up and Down arrow buttons or entering directly numeric values in the text part. This area displays nothing when multiple image or pallet rectangles are selected.

Figure 3-10: Select List Area



The Select list area displays a list of images and pallets which can be dealt with. This list allows you to select image or pallet rectangles. Rectangle which was selected in the VRAM image area is regarded to be selected also in the list. Use of the Shift or Ctrl key allows you to select multiple rectangles. Dragging also allows selection of multiple rectangles.

Figure 3-11: Menu Bar



Sorting menu

- **Texture**
Sets the left top corner of all textures to a side of the nearest texture area or display range. Selection of this menu after setting roughly textures with the mouse arranges them orderly.
- **Pallet**
Paves all pallets vertically from the bottom of a range which is currently displayed.

Grid menu

- **Non**
Disables grid function. Texture is located to coordinates you specified with the mouse.
- **Texture page**
Moves and lines up textures along the nearest texture page borderline from coordinates you specified with the mouse.

Both X and Y coordinates are lined up along the texture page borderline if you selected "XY", only X coordinates are lined up if you selected "X" and only Y coordinates are lined up if you selected "Y".
- **Magnet**
Moves and sets textures to a side of the nearest texture area or display range from coordinates you specified with the mouse. Both X and Y coordinates are lined up along the texture page borderline if you selected "XY", only X coordinates are lined up if you selected "X" and only Y coordinates are lined up if you selected "Y".

Zoom menu

Sets scales for the image display in the "VRAM image area". Use this for small texture, pallet and others when it is difficult to select or move them with ordinary scales.

Display range menu

Changes display resolutions which are actually used on PlayStation. Change of the display range changes the range of the display area.

Press "OK" button to reflect edited VRAM images on the Parameter Setting window. Selection of "Cancel" button or "Close" of the system menu invalidates move of image or pallet.

Remarks

The compressed format BMP, JPEG compressed PICT, 32-bit PICT and PICT without bit map cannot be read in.

Write-out as the compressed format BMP or JPEG compressed format PICT file is not allowed.

Write-out file bit depth is limited to 4, 8, 16 or 24-bit.

In data conversion in which the bit depth reduces, for example, from 16-bit to 8-bit, color information is approximated. As this approximation is not realized with a method such as color compression but with color map on which R, G and B are uniformly allocated, the color accuracy may deteriorate.

As a rule, overwriting to the file which was read in is not allowed. However, overwriting is allowed when both read-in format and write-out format are the TIM and only image origin and pallet origin were changed. This function is equal to TIMPOS.EXE of Commandline version.

TIMEXP: Plug-in for use with Photoshop 2.5 and 3.0

Version 1.2

Documentation date: 11/18/97

Description

Features

TIMEXP.8BE is a plug-in module which allows you to output image which is created and amended with Adobe Photoshop to DTL-H201A and view it on a video monitor. With this, you can check how the image data is displayed on the PlayStation system.

This software has the following features:

- Supports 4 color modes
The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that the PlayStation system can deal with.
- Supports 8 display modes
The software supports all display modes that the PlayStation system can deal with. Change of the VRAM display start address allows you to display any VRAM area.
- Allows you to specify the VRAM location
The software allows you to load the Display area, Texture page and Texture CLUT to any VRAM address. With this, you can also turn on/off Texture mapping to the Display area.
- Supports the transparent control bit (partly)
The software provides you with a function to use the transparent control bit (STP) unique to the PlayStation system.

Operating requirement

TIMEXP.8BE is subject to the Adobe Photoshop operating requirement because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar to its operating procedure. For details of Adobe Photoshop, refer to instruction manuals and others attached to Adobe Photoshop.

System configuration

In order to use TIMEXP.8BE, the following software is required:

Adobe Photoshop 2.5J or 3.0

Preparation

Installation of program

For use with *Adobe Photoshop 2.5*: Copy TIMEXP.8BE at C:\PSXGRAPH\BIN to PLUGINS subdirectory in a directory in which PHOTOSHP.EXE was installed.

For use with *Adobe Photoshop 3.0*: Copy into subdirectory PLUGINS\EXPORT.

Setup of operating environment

Start up ABOARD.EXE, a tool for setting the DTL-H201A board address and set the DTL-H201A address.

Check of installation and setup

Starting up Adobe Photoshop and selecting "HELP/ABOUT PLUG-IN/TIM DISPLAY..." option from the Menu bar to open the About box allows you to check the address which is set. (The default address is 0x1340.)

Operations

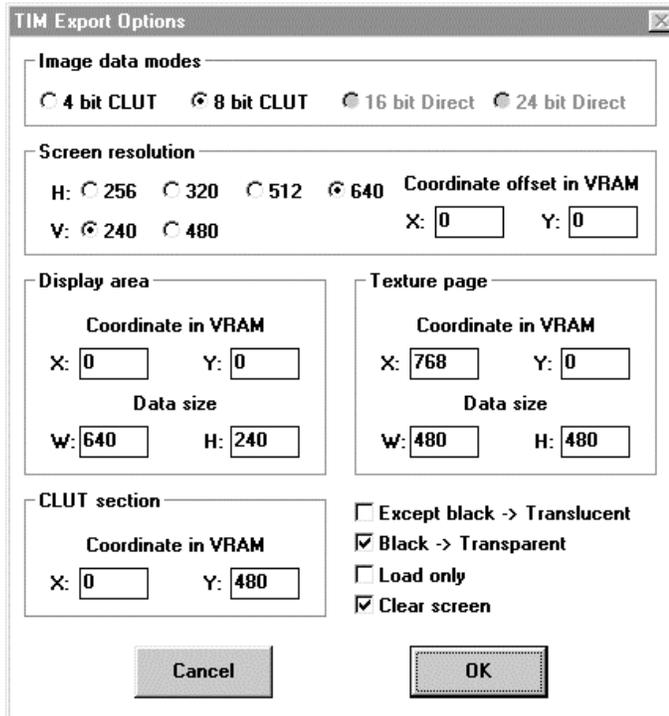
In order to view the Adobe Photoshop image on DTL-H201A

Select a window you want to access and then output the Adobe Photoshop image information to DTL-H201A with "TIM DISPLAY..." command to view it on the display.

How to use "TIM DISPLAY..." command

1. Select "OUTPUT PLUG/TIM DISPLAY..." option from the file menu to access the TIM Export Options dialog box.

Figure 3-12: TIM Export Options Dialog Box



2. Set or select any desired value for the following settings and click "OK".

- **Color mode**

Set a DTL-H201A pixel format. One of the following formats is available, depending on the Adobe Photoshop image mode.

Index color → 4 or 8-bit CLUT

RGB color → 16 or 24-bit Direct

- **Display mode**

Set the DTL-H201A horizontal (W) and vertical (H) resolutions. Change of the start address allows you to begin with any VRAM address you want to display (left top corner)

- **Display area**

Set a rectangular VRAM address (left top corner) to map the texture data and the width and height. This is invalid in Load only mode.

- **Texture page**

Set a VRAM address (left top corner) to store the texture data and the width and height. In Load only mode, the address is automatically set to (0,0).

- **Texture CLUT**

Set a VRAM address (left top corner) to place the CLUT. This is invalid when the color mode is 16/24-bit.

- **Letting any other color than the black turn semitransparent**
In order to let a color turn semitransparent, set an Adobe Photoshop format pixel which has a value of 8 or more for at least one of R, G and B or transparent control bit which is CULT entry.
- **Letting the black turn transparent**
In order to let the black turn transparent, clear any Adobe Photoshop format pixel which has values of 7 or less for all of R, G and B or transparent control bit which is CULT entry.
- **Load only**
Changes the system mode to Load only mode. This mode merely sends the texture data to the VRAM. It does not map it into a rectangle of display area. Therefore, the transparent control is invalid. Selection of the 24-bit color mode sets automatically this mode.
- **Screen clear**
Clears the whole VRAM before drawing.
- **Cancel**
Cancels drawing.
- **OK**
Outputs the drawing command to DTL-H201A.

Note

In the Adobe Photoshop format, components R, G and B have 8-bit resolution. On the other hand, each resolution of DTL-H201A is 5-bit, except 24-bit mode. Therefore, outputted to DTL-H201A, 5 high order bits are valid but 3 low order bits are ignored unconditionally.

For example:

Adobe Photoshop		DTL-H201A internal notation
R, G, B):(0x08, 0x07, 0xff)	→	[0x01, 0x00, 0x1f]

DTL-H201A is provided with the transparent control bit (except 24-bit mode) but Adobe Photoshop has nothing which corresponds to it. This module is provided with 2 types of check box to deal with the transparent control bit. These boxes allow you to use the transparent control bit to some degree. However, "transparent" and "(semi) nontransparent black" cannot be mapped into a display area at a time. In such case, use (R, G, B)=(0, 0, 8) and others instead of "black".

The transparent control bit is valid only when texture is mapped into the display area.

TIMexpE: Plug-in for use with Photoshop 4.0, Windows or Artist Board

Version 4.0E

Documentation date: 8/29/97

Description

Features

TIMexpE.8BE is a plug-in module for Adobe Photoshop 4.0 which allows you to output an image which is created and modified in Adobe Photoshop 4.0 to the Artist Board and view it on a video monitor. With this, you can check how the image data is displayed on the PlayStation system.

This software has the following features:

- Supports 4 color modes
The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that the PlayStation system deals with.
- Supports 8 display modes
The software supports all display modes that the PlayStation system deals with. Change of the VRAM display start address allows you to display any VRAM area.
- Allows you to specify the VRAM location
The software allows you to load the Display area, Texture page and Texture CLUT to any VRAM address. With this, you can also turn on/off Texture mapping to the Display area.

The software provides you with a function to use the transparent control bit (STP) unique to the PlayStation system.

Operating requirement

TIMexpE.8BE is subject to Adobe Photoshop operating requirements because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar with its operating procedures. For details about Adobe Photoshop, refer to instruction manuals and other information attached to Adobe Photoshop.

System configuration

TIMexpE.8BE requires the following operating environment:

Adobe Photoshop 4.0, Windows 95, PlayStation Artist Board

Preparation

Installation of program

Copy **TIMexpE.8BE** into the PLUGINS\EXPORT subdirectory of the Photoshop directory. Copy the initialization file **TIMexp.ini** into the system directory C:\Windows.

Setup of operating environment

Start up ABOARD.EXE, a tool for setting the Artist Board address, and set the address.

Setup of initialization file "TIMexp.ini"

This file is located in the system directory and includes some program settings as below:

Default values

Whole default values of the Export Options dialog box can be customized by editing the [DefaultValue] section. See Table 3-1 for more details.

Record masks

When you record actions, plug-ins will output whole parameters. Although this export plug-in contains 26 parameters, a few parameters may be changed in most cases. Setting a parameter at 1 in the [RecordMask] section allows you to disable a parameter output which you did not change and therefore do not have to notice.

In order to record actions with more flexibility, use fewer parameters. The record mask is a useful function for reducing the number of parameters used to a minimum.

The values of maskable parameters are almost the same as the default values. The exception "Base" is a non-maskable parameter. See the initial file for more details.

List of saved values

A list of whole parameter values is needed to export an image. When you click the "Save" button in the Export Options dialog box, the plug-in will immediately generate / update the list associated with the full path of the target image file.

The list consists of 26 parameters in the following order:

- Color,
- DispModeH, DispModeV, DispStartX, DispStartY,
- DispAreaX, DispAreaY, DispAreaW, DispAreaH,
- TexX, TexY, TexW, TexH,
- ClutX, ClutY,
- ABR, Lower3, StpOthers, StpBlack, LoadOnly, ClearScreen,
- ColorAuto, TexAddrAuto, TexSizeAuto, ClutAddrAuto, StpBitAuto

Table 3-1: Parameters specifiable in the [DefaultValue] section of TIMexp.ini

Entry name	Content
Base(0)	Base setting (0: Default values, 1: Previously saved values)
ColorAuto(0)	Automatic Color mode (0: Off, 1: Auto)
Indexed(1)	Color mode for Indexed color (0: 4 bit CLUT, 1: 8 bit CLUT)
Direct(2)	Color mode for RGB color (2: 16 bit Direct, 3: 24 bit Direct)
DispModeH(0)	Horizontal screen resolution (0: 256, 1: 320, 2: 512, 3: 640)
DispModeV(0)	Vertical screen resolution (0: 240, 1: 480)
DispStartX(0)	Display start address: X
DispStartY(0)	Display start address: Y
DispAreaX(0)	X Address of display area
DispAreaY(0)	Y Address of display area
DispAreaW(1)	Width of display area
DispAreaH(1)	Height of display area
TexAddrAuto(0)	Automatic texture address (0: Off, 1: Auto)
TexX(0)	X Address of texture
TexY(0)	Y Address of texture
TexSizeAuto(1)	Automatic texture size (0: Off, 1: Auto)
TexW(1)	Width of texture size
TexH(1)	Height of texture size
ClutAddrAuto(0)	Automatic CLUT address (0: Off, 1: Auto)
ClutX(0)	X Address of CLUT
ClutY(0)	Y Address of CLUT
ABR(0)	Translucent rate (0 is compatible with TIMEXP version 1.2E) 0: Foreground: 50% + Background: 50% 1: Foreground: 100% + Background:100% 2: Foreground:-100% + Background:100% 3: Foreground: 25% + Background:100%
Lower3(0)	Rounding method (0 is compatible with TIMEXP version 1.2E) 0: round down, 1: round off, 2: round up
StpBitAuto(0)	Automatic STP bit control (0: Off, 1: Auto)
StpOthers(0)	Except black -> Translucent (0: Off, 1: On)
StpBlack(0)	Black -> Transparent (0: Off, 1: On)
LoadOnly(0)	Load Only (0: Off, 1: On)
ClearScreen(0)	Clear screen initially (0: Off, 1: On)
WnNotSaved(1)	Warn of "Saved values not found" (0: Silent, 1: Warn)
WnModeDisagree(1)	Warn of "Color mode disagreement" (0: Silent, 1: Warn)
ReadIniFile(0)	Read initial file (0: Once at initial call, 1: At every call)

Caution: An entry name prefixed with ';' is ignored and the plug-in adopts the value in parentheses.

How the TIM format plugin works with the TIM export plug-in

When the TIM format plug-in **TIMfmtE.8bi** opens a TIM file, it always scans the file header and the contents of the CLUT and texture image data in order to check on STP bit, and writes the result into Caption section of Photoshop's *File Information* data field. If the Caption is valid and the color mode agrees with the target image, then the export plug-in can enable four "Auto" check-boxes: "Color mode", "Texture Page address", "Texture CLUT address" and "STP bit". These check-boxes will be disabled when the Caption is incorrect.

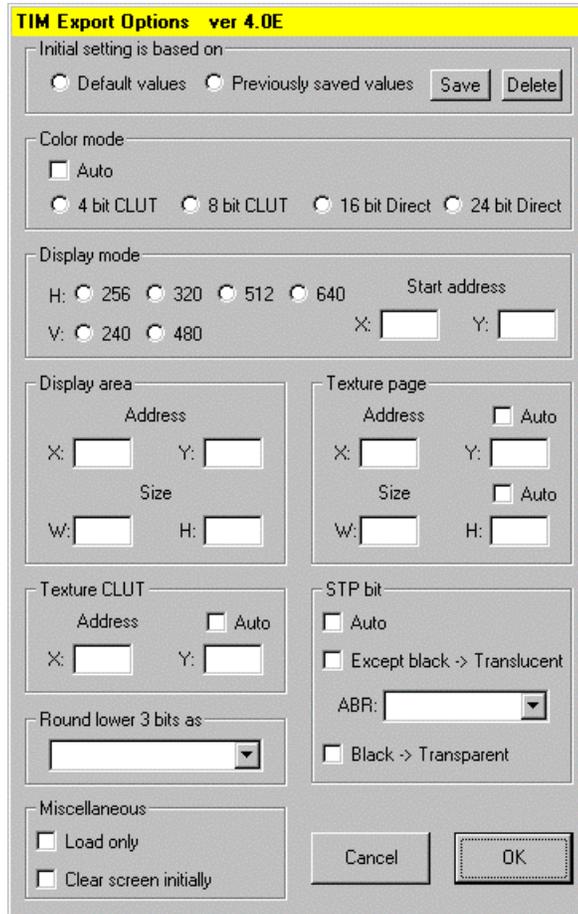
Operations

To export an image to the Artist Board

Choose File > Export > PlayStation TIM export.

The TIM Export Options dialog box is displayed as below.

Figure 3-13: TIM Export Options Dialog Box



- **Initial setting is based on**

Selects the base of the initial setting values of the Export Options dialog box: either “Default values” or “Previously saved values”.

**Default values:* Retrieve from [DefaultValue] section of the initial file.

**Previously saved values:* Retrieve the saved values which was saved as follows.

The "Save" push button allows you to save / update the whole parameters displayed in the dialog box into the initial file associated with the full path of the target image file immediately. Do it with some care because you may not cancel this action.

The "Delete" push button allows you to delete the saved values list in the initial file. After deletion of the list, you may select “Previously saved values”, if you wish, until the dialog box is closed. Do it with some care because you may not cancel this action.

Caution: A plug-in will always read a descriptor (*) passed from Photoshop and update the specified parameters before opens the dialog box. The values in the dialog box will not necessarily MATCH with the values selected at “Initial setting based on” because some parameters may be changed by the descriptor passed from Photoshop.

When you set all parameters in the [RecordMask] section and record an export action command without changing any parameters, the plug-in will record a command with "Based on" parameters only. This command allows you to export an image with the original settings as you specified.

*: A descriptor is a component of a command for an action, and is a pair of data values in the form of [key value] that describe a property of an object or the parameters of an event.

- **Color mode**

Sets the pixel format to be used with the Artist Board. Choose one of the following settings according to the image mode:

Indexed color: 4 bit CLUT or 8 bit CLUT

RGB color: 16 bit Direct or 24 bit Direct

- **Display mode**

Sets the horizontal (H) and vertical (V) resolution of the Artist Board screen. By entering a start address, you can see any area of VRAM.

- **Display area**

Sets the address in VRAM (from the upper left corner) and the width and height of the rectangle to be texture-mapped. These are disabled in the Load Only mode.

- **Texture page**

Sets the address in VRAM (from the upper left corner) and the width and height of the texture data. The width and height provided by Photoshop are available by clicking "Auto".

The address is automatically set to (0, 0) in the Load Only mode.

- **Texture CLUT**

Sets the address in VRAM (from the upper left corner) of the CLUT. These parameters are disabled when the color mode is set to 16 bit Direct or 24 bit Direct.

- **STP bit**

STP bit settings are enabled except in the Load Only mode.

* *Except black -> Translucent*

Sets the transparency control bit for pixels or CLUT entries, other than those where the (R, G, B) values are set to (0, 0, 0) after 8 bit to 5 bit conversion. As a result, all colors other than black become translucent only in translucent mode.

* *ABR*

When you check the check box of "Except black -> Translucent", you can select the rate of translucence as below:

1) Foreground: 50% + Background: 50% (compatible with version 1.2E)

2) Foreground: 100% + Background: 100%

3) Foreground: -100% + Background: 100%

4) Foreground: 25% + Background: 100%

* *Black -> Transparent*

Clears the transparency control bit for those pixels or CLUT entries where the (R, G, B) values are set to (0, 0, 0) after 8 bit to 5 bit conversion. As a result, black becomes transparent, regardless of translucency mode.

- **Round lower 3 bits as**

In Photoshop, RGB components are represented in a resolution of 8 bits for each color, but the Artist Board supports only 5 bits for each (except in the 24 bit Direct mode). Therefore, a transformation from 8 bits to 5 bits is required when exporting image data. This export plug-in lets you select the transformation method from the options given here:

- 1) 00000.XXX > 00000 round down unconditionally (compatible with TIMEXP version 1.2E)
- 2) 00000.1XX > 00001 round up if Bit2 is 1 (here Bit0 is the LSB)
- 3) 00000.XXX > 00001 round up unconditionally

- **Miscellaneous**

- * *Load Only*

Sets the Load Only mode. In this mode, texture data only is sent to VRAM, and mapping of the display area is not performed. Transparency control bits are therefore ignored. This mode is automatically selected when the color mode is set to 24 bit Direct.

- * *Clear screen initially*

Clears VRAM before exporting image data.

- **Cancel**

Cancels drawing and keeps parameter values unchanged for recording an action again.

- **OK**

Exports image data to the Artist Board.

Notes

A method of setting the contents displayed in Export Options dialog box

The contents displayed in the Export Options dialog box are set as shown below:

1. If you set "Base=1" (previously saved values) in the [DefaultValue] section of the initial file, then this plug-in will attempt to find the saved values list and set them. The default values will be used if the search fails, or there is color mode disagreement (*) occurs, or "Base=0" is set.
 - *: The color mode specified in the descriptor or the saved values list disagrees with the color mode of the target image data.
2. If Photoshop passes the plug-in a descriptor, continue reading below. Otherwise go to 4.

First, the plug-in will check the color mode. If the color mode specified in the descriptor agrees with the target image data, then the plug-in changes the parameters specified in the descriptor. Otherwise the descriptor will be discarded.
3. If the descriptor sets "Base=1", then the plug-in attempts to find the list of saved values and set them. The default values will be used if the search fails or the color mode disagreement occurs, or "Base=0" is set.
4. "Auto" check-boxes will become effective at this stage, if possible.
5. The dialog box will be opened.

Caution: When this plug-in finishes exporting an image normally, whole parameters are passed to Photoshop as a descriptor. Photoshop keeps this descriptor, and passes it back to this plug-in on re-invocation. Therefore, some warnings below may appear when receiving a preceding descriptor.

- * Color mode disagreement in the correct operations:

When you export an RGB color image after exporting another indexed color image with "Color mode: 8 bit CLUT", on the condition that the plug-in outputs the color mode descriptor by setting "Color=0" in the [RecordMask] section of the initial file, Photoshop passes a descriptor of "Color mode: 8 bit CLUT" to the plug-in. In this case, a color mode disagreement will occur.

- * Unexpected changes of parameters:

For example, on the condition of "Base = 0 (Default values)" in the initial file, when you export an image without saved values after exporting another image with "Based on: saved values", the plug-in will warn you "No saved values found", because Photoshop passes the plug-in a descriptor of "Based on: saved values".

No descriptor will be passed to Photoshop if you cancel an export operation.

Important Points

Transparency control bits are only valid in the display area where texture mapping is drawn.

You cannot perform mapping in a single display area using both "Transparent" and "Translucent except black" attributes at the same time. For situations such as this, set all "black" pixels to an alternate value, such as (R, G, B) = (0, 0, 1), in the TIM format.

TIMfmtE: Plug-in for use with Photoshop 4.0, Windows 95 or NT

Version 4.0(E)

Documentation date: 8/29/97

Description

TIMfmtE.8BI is a plug-in module for Adobe Photoshop 4.0, which allows you to import directly into Photoshop two dimensional TIM format files that the PlayStation system deals with, and to output images created with Photoshop into TIM format files.

This software has the following features:

- Supports 4 TIM format modes
The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that PlayStation system deals with.
- Supports the transparency control bit
The software allows you to use the transparent control bit (STP) to save the image data as a TIM format file.
- Allows you to specify the VRAM load address
The software allows you to specify Texture CLUT and Texture page addresses to load image data, saved as a TIM format file, to the VRAM.

Operating requirement

TIMFMT.8BI is subject to the Adobe Photoshop operating requirement because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar to its operating procedure. For details of Adobe Photoshop, refer to instruction manuals and other information attached to Adobe Photoshop.

System configuration

TIMfmtE.8BI requires the following operating environment.

Adobe Photoshop 4.0, Windows 95 or NT

Preparation

Installation of program

Copy **TIMfmtE.8BE** into the PLUGINS\FORMAT subdirectory of the Photoshop directory. Copy the initialization file **TIMfmt.ini** into the system directory C:\Windows.

Setup of initialization file "TIMfmt.ini"

This file is located in the system directory and includes some program settings as below:

Default values

When you save a new image or one from another format as a TIM file, there is no File Information to be referred to. In this case, this plug-in will refer to the [DefaultValue] section of the initial file. See Table 3-2 for more details.

Record masks

When you record actions, plug-ins will output whole parameters. Although this file format plug-in contains 15 parameters, a few parameters may be changed in most cases. Setting a parameter at 1 in the [RecordMask] section allows you to disable a parameter output which you did not change and therefore do not have to notice.

In order to record actions with more flexibility, use fewer parameters. The record mask is a useful function for reducing the number of parameters used to a minimum.

The values of maskable parameters are almost the same as the default values. See the initial file for more details.

Table 3-2: Parameters specifiable in the [DefaultValue] section of TIMfmt.ini

Entry name	Content
ColorAuto(0)	Automatic Color mode (0: Off, 1: Auto)
Indexed(1)	Color mode for Indexed color (0: 4 bit CLUT, 1: 8 bit CLUT)
Direct(2)	Color mode for RGB color (2: 16 bit Direct, 3: 24 bit Direct)
TexAddrAuto(0)	Automatic texture address (0: Off, 1: Auto)
TexX(0)	X Address of texture
TexY(0)	Y Address of texture
TexSizeAuto(1)	Automatic texture size (0: Off, 1: Auto)
TexW(1)	Width of texture size
TexH(1)	Height of texture size
ClutAddrAuto(0)	Automatic CLUT address (0: Off, 1: Auto)
ClutX(0)	X Address of CLUT
ClutY(0)	Y Address of CLUT
Lower3(0)	Rounding method (0 is compatible with TIMFMT version 1.2E) 0: round down, 1: round off, 2: round up
StpBitAuto(0)	Automatic STP bit control (0: Off, 1: Auto)
StpOthers(0)	Except black -> Translucent (0: Off, 1: On)
StpBlack(0)	Black -> Transparent(0: Off, 1: On)
WnModeDisagree(1)	Warn of "Color mode disagreement" (0: Silent, 1: Warn)
ReadIniFile(0)	Read initial file (0: Once at initial call, 1: At every call)

Caution: An entry name prefixed with ';' is ignored and the plug-in adopts the value in parentheses.

Reference to File Information

A non-image data field, *File Information*, has been introduced into Photoshop for versions 3.0 onward. When you open a file with this plug-in, it scans the file header and the contents of the CLUT and texture data in order to check on the STP bit, and writes the results into the Caption section of the *File Information* data field.

If the Caption is valid, then this plug-in can enable four "Auto" check-boxes: "Color mode", "CLUT address", "Texture address", and "STP bit" in the TIM Format Output Options dialog box. Original values may be set by choosing "Auto".

Some auto check-boxes will be disabled if the Caption does not exist or is inadequate.

You can look at the contents of the *File Information* data field by choosing File > File Info from the menu bar.

Caution: When you save a image with this plug-in, it will **always** update the *File Information* of the target window. Suppose you made a copy from *A.TIM* to *Acopy.TIM* by selecting File > Save a Copy, and changed the CLUT address of *Acopy.TIM*. The CLUT address in the *File Information* data field of the target window, *A.TIM*, would be changed to the value put into *Acopy.TIM*, though the target window's name would be still be *A.TIM*, and *not change* to *Acopy.TIM*.

Operations

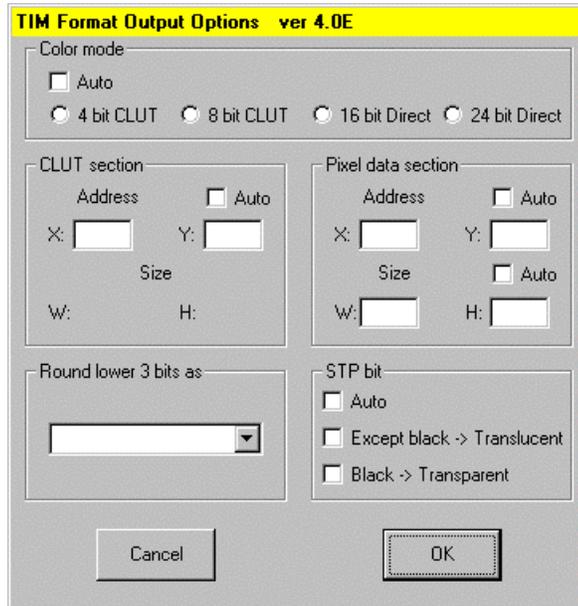
To open a TIM file

Choose File > Open, and select a TIM format file.

To save a file in TIM file format

1. In the "Save" or "Save As" dialog box, select " PlayStation TIM (*.TIM)" from the "File Format" list.
2. Type a file name, and click "Save".
3. The TIM Format Output Options dialog box is displayed as below:

Figure 3-14: TIM Format Output Options Dialog Box



- **Color mode**
Choose one of the following settings according to the color mode of the image:
Indexed color: 4 bit CLUT or 8 bit CLUT
RGB color: 16 bit Direct or 24 bit Direct
- **CLUT section**
Sets the address in VRAM (from the upper left corner) of the CLUT. The size fields are filled in automatically.
- **Pixel data section**
Sets the address in VRAM (from the upper left corner) and the width and height of the texture image data. The sizes provided by Photoshop are usually available, and are used by clicking "Auto".
- **Round lower 3 bits as**
In Photoshop, RGB components are represented in a resolution of 8 bits for each color, but TIM format supports only 5 bits for each (except in the 24 bit Direct mode). Therefore, a transformation from 8 bits to 5 bits is required when saving image data. This format plug-in lets you select the transformation method from the options given here:
 - 1) 00000.XXX > 00000 round down unconditionally (compatible with TIMFMT version 1.2E)
 - 2) 00000.1XX > 00001 round up if Bit2 is 1 (here Bit0 is the LSB)
 - 3) 00000.XXX > 00001 round up unconditionally

- **STP bit**
STP bit settings are enabled, except in 24 bit Direct mode.
** Except black -> Translucent*
Sets the transparency control bit for pixels or CLUT entries, other than those where the (R, G, B) values are set to (0, 0, 0) after 8 bit to 5 bit conversion. As a result, all colors other than black become translucent only in translucent mode.
** Black -> Transparent*
Clears the transparency control bit for those pixels or CLUT entries where the (R, G, B) values are set to (0, 0, 0) after 8 bit to 5 bit conversion. As a result, black becomes transparent, regardless of translucency mode.
- **Cancel**
Cancels the save operation and keeps the values the same on recording an action again.
Suppose you save NEW.TIM with the Save As command over OLD.TIM and respond "Yes" to the message that warned you about replacing an existing file. No change will be made to OLD.TIM by choosing "Cancel" at this step.
- **OK**
Saves the image data in TIM file format.

Notes

A method of setting the contents displayed in Output Options dialog box

The contents displayed in the Format Output Options dialog box are set as shown below:

1. The default values in the initial file are adopted as initial setting values.
2. If Photoshop passes the plug-in a descriptor (*1), continue reading below. Otherwise go to 3.
First, the plug-in will check the color mode. If a color mode disagreement (*2) has occurred, then the plug-in will discard the descriptor. Otherwise the plug-in changes the parameters specified in the descriptor.
**1: A descriptor is a component of a command for an Action, and is a pair of data values in the form of [key value] that describe the property of an object or the parameters of an event.*
**2: The color mode specified in the descriptor disagrees with the color mode of target image data.*
3. "Auto" check-boxes will become effective at this stage, if possible.
4. The dialog box will be opened.

Caution: When this plug-in finishes saving a file normally, whole parameters are passed to Photoshop as a descriptor. Photoshop keeps this descriptor, and passes it back to this plug-in on re-invocation. Therefore, some warnings below may appear when receiving a preceding descriptor.

For example, when you save an RGB color image after saving another indexed color image with "Color mode: 8 bit CLUT", on the condition that the plug-in outputs the color mode descriptor by setting "Color=0" in the [RecordMask] section of the initial file, Photoshop passes a descriptor of "Color mode: 8 bit CLUT" to this plug-in. In this case, a color mode disagreement will occur.

No descriptor will be handed to Photoshop if you cancel the save operation.

Important Points

You cannot save a single TIM file which contains both "transparent" and "translucent/non-transparent black" at the same time. For situations such as this, set all "translucent/non-transparent black" pixels to an alternate value such as (R, G, B) = (0, 0, 1) in the TIM format.

3-36 2D Graphics Conversion Utilities

RGB components in Photoshop are used at a resolution of 8 bits each, but the TIM format only uses 5 bits for each (except in 24 bit Direct mode). Therefore, when a TIM file is opened, 0s will be inserted into the lower 3 bits.

Photoshop's indexed color images use a 256-entry CLUT, so even for 4-bit CLUT, TIM files of up to 256 colors may be maintained. However, when the image is displayed in Photoshop, only the first 16 colors will be used.

TIMFMT/TIMformatE: Plug-ins for use with Photoshop 2.5 and 3.0

Version 1.2 of TIMFMT.8BI

Version 1.3 of TIMformatE.8BI

Documentation date: 11/18/97

Description

These modules allow you to import directly two dimensional TIM format files that the PlayStation system deals with into Adobe Photoshop and output images created with Adobe Photoshop into the TIM format file.

TIMFMT.8BI is a plug-in module for use with Adobe Photoshop 2.5.

TIMformatE.8BI is for use with Photoshop 3.0.

This software has the following features:

- Supports 4 TIM format modes
The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that PlayStation system can deal with.
- Supports the transparent control bit (partly)
The software allows you to use the transparent control bit (STP) to save the image data as the TIM format file.
- Allows you to specify the VRAM load address
The software allows you to specify Texture CLUT and Texture page addresses to load the image data which was saved as the TIM format file to the VRAM.

Operating requirement

The plug-in is subject to the Adobe Photoshop operating requirement because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar to its operating procedure. For details of Adobe Photoshop, refer to instruction manuals and others attached to Adobe Photoshop.

Preparation

Installation of program

Adobe Photoshop 2.5 users:

Copy TIMFMT.8BI to PLUGINS subdirectory in the directory in which PHOTOSHP.EXE is installed.

Adobe Photoshop 3.0 users:

Copy TIMformatE.8BI into the subdirectory PLUGINS\FORMATS.

Operations

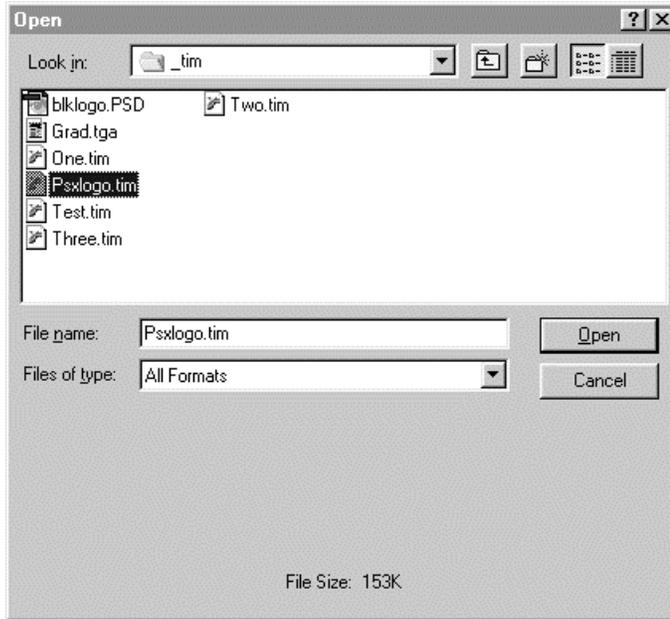
In order to open a TIM format file as an Adobe Photoshop document

"Open..." command is available to open a document when the file extension of "*.TIM" is attached to a TIM format file you want to open. Otherwise, use "Open in specified format..." command unless the file extension of "*.TIM" is attached to.

How to use "Open..." command

1. Select "Open..." option from the file menu to access "Open" dialog box.
2. Select a TIM format file you want to open and click "OK".

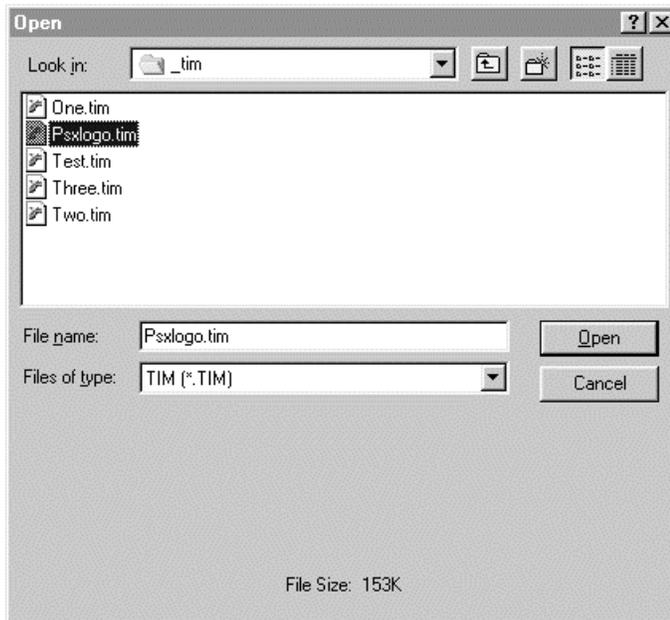
Figure 3-15: "Open" Dialog Box



How to use "Open in specified format..." command

1. Select "Open in specified format..." option from the file menu to access "Open in specified format" dialog box.
2. Select "TIM (*.TIM)" option from the "File format" drop-down list.
3. Select a TIM format file you want to open and click "OK".

Figure 3-16: "Open in Specified Format" Dialog Box



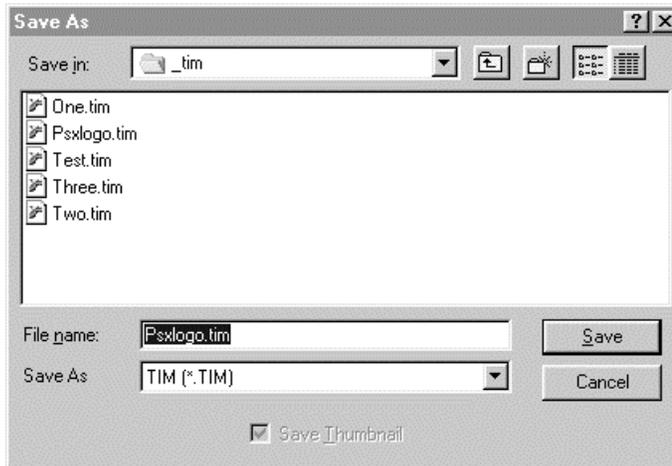
In order to save the Adobe Photoshop image as the TIM format file

In order to save a file which has any other format than the TIM format as the TIM format file, use "Save with another name..." command.

How to use "Save with another name..." command

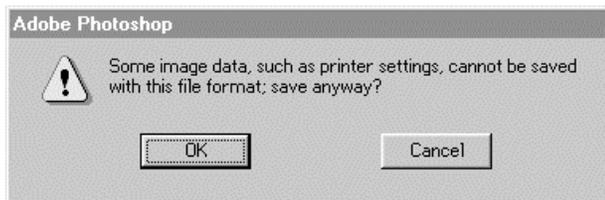
1. Select "Save with another name..." option from the file menu to access "Save with another name" dialog box.
2. Select "TIM (*.TIM)" option from the "File format to save" drop-down list.
3. Enter any desired filename and click "OK".

Figure 3-17: "Save with another Name" Dialog Box



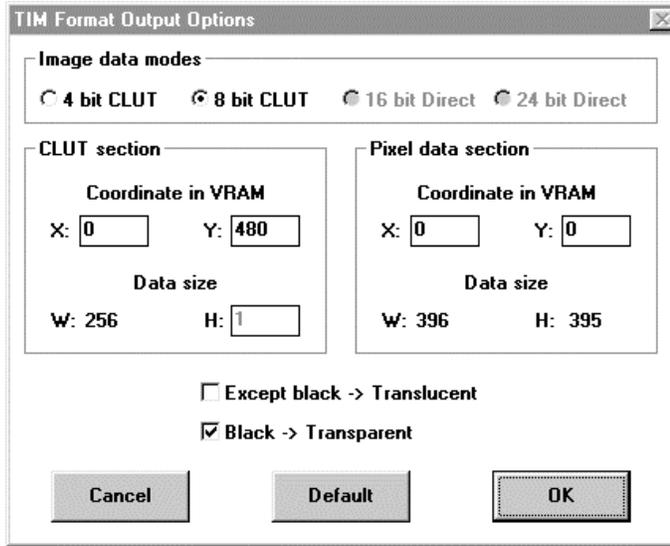
4. A warning "Image information which cannot be saved exists. Save?" appears. Click "OK".

Figure 3-18: Warning about Image Information which cannot be Saved



5. TIM Format Output Options dialog box appears. The Texture CLUT address and Texture page address show values of a TIM format file which was opened (or saved) at the previous time.

Figure 3-19: TIM Format Output Options dialog box



6. Set or select any desired value for the following settings and click "OK".

- **Image data mode**

Set PMODE of the TIM format file. One of the following formats is available, depending on the type of original image.

Index color → 4 or 8-bit CLUT
 RGB color → 16 or 24-bit Direct

- **CLUT section**

Set a VRAM CLUT address (left top corner). You can set the width (W) and height (H) of CLUT as follows:

4-bit CLUT → Width: 16 (fixed), Height: 1-16 (as necessary)
 8-bit CLUT → Width: 256 (fixed), Height: 1 (fixed)

- **Pixel data section**

Set a VRAM pixel data address (left top corner). The width (W) and height (H) of image are shown but unchangeable.

- **Letting any other color than the black turn semitransparent**

In order to let a color turn semitransparent only in semitransparent mode, set an Adobe Photoshop format pixel which has a value of 8 or more for at least one of R, G and B or transparent control bit which is CULT entry. (The default turns this check box OFF.)

- **Letting the black turn transparent**

In order to let the black turn always transparent regardless of semitransparent mode, clear any Adobe Photoshop format pixel which has values of 7 or less for all of R, G and B or transparent control bit which is CULT entry. (The default turns this check box ON.)

- **Default**

Sets standard values for the CLUT section address and pixel section address. Undo of this operation is not allowed.

- **Cancel**

Cancels saving.

- **OK**

Saves the image information as the TIM format file.

Note

In Adobe Photoshop, components, R, G and B have the 8-bit resolution. On the other hand, each TIM format resolution is 5-bit, except 24-bit mode. Therefore, to say strictly, equivalent conversion from Adobe Photoshop to the TIM format file is not realized. (The 24-bit mode realizes it.) This module adds '0's for 3 low order bits when it opens the TIM format file and picks up 5 high order bits but ignores 3 low order bits when it saves the file.

For example

Adobe Photoshop TIM format file Adobe Photoshop
 (R,G,B):(0x08,0x07,0xff) → (0x01,0x00,0x1f) → (0x08,0x00,0xf8)

The TIM format is provided with the transparent control bit, except 24-bit mode but Adobe Photoshop has nothing which corresponds to it. Therefore, this module deals with the transparent control bit as follows:

When this plug-in opens the TIM file

All transparency bits are lost. This leads both "transparent" and "(semi) nontransparent black" to the same Adobe Photoshop value, (R, G, B)=(0, 0, 0).

When it saves a file as the TIM format file

Two types of check box are provided to deal with the transparency control bit. These boxes allow you to use the transparency control bit to some degree. However, you cannot store "transparent" and "(semi) nontransparent black" in a TIM format file at the same time. In such cases, use (R, G, B)=(0, 0, 8) (equal to [0, 0, 1] of the TIM file) for each "black" pixel.

The Adobe Photoshop indexed color image always holds the 256-color CLUT. For the 4-bit mode TIM format file, input and output of the CLUT of up to 256 colors are available. However, Adobe Photoshop uses only the first 16 colors to display it.

MGAME.ADDIN: Plug-in for Alias Wavefront

The PlayStation game console translator supports:

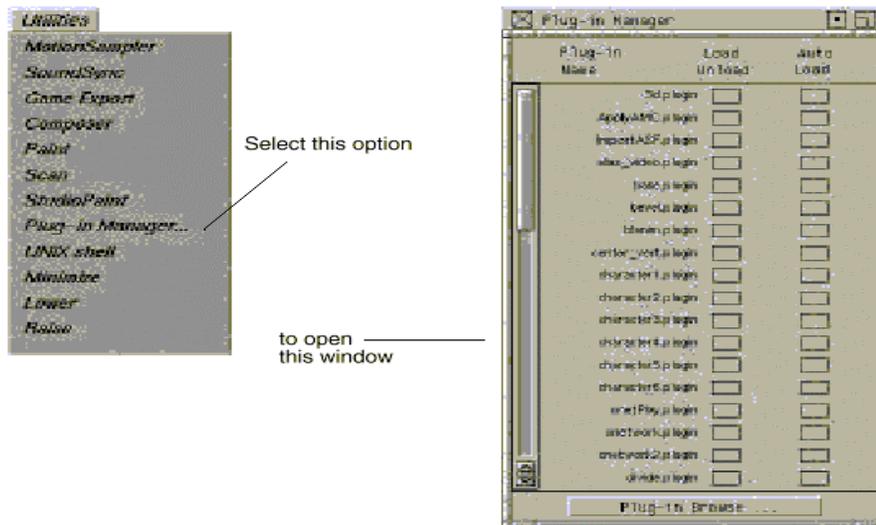
- Hierarchical animation
- Support of per-poly shaders and per-poly textures
- Output of animation separately from modeling data
- Optimized output of TOD data (removes Identity TRS components)
- MetaCycle Preview on PlayStation
- Output of VDF/NDF files for MIME animation
- Output of sequences of TMD files
- Control of STP bit in TIM files

How to Use

You load the PlayStation development system from the Make Game translator window, which means you first have to load the mgame.addin plug-in.

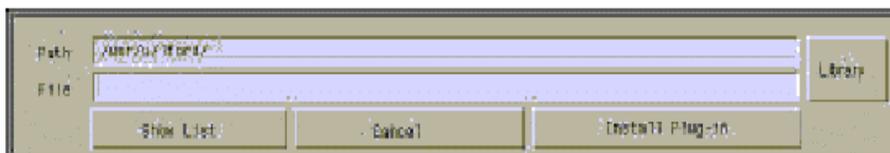
1. Select Plug-in Manager from the Utilities menu to open the Plug-in Manager window.

Figure 3-20: Opening the Plug-in Manager window



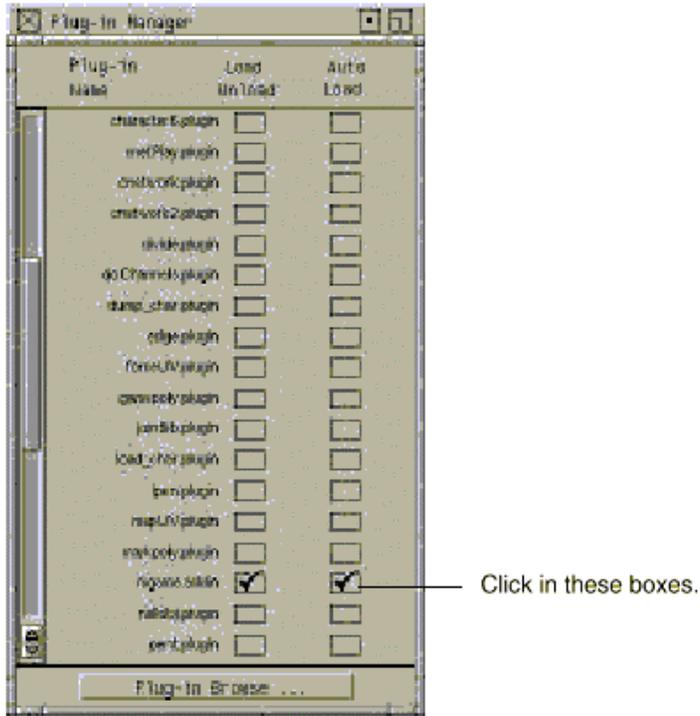
2. Select mgame.addin from the list if the \$GAMES_LOCATION/Games/plugins is in the default plugin search path. If mgame.addin is not in the list, click the Plug-in Browse... button at the bottom of the Plug-in Manager window to open the File Requestor.

Figure 3-21: Search path and file name window



3. Click Show List to display a list of available plug-ins and select the mgame.addin icon from the File Lister, or type mgame.addin in the File box.
4. Click the Install Plug-in button.
5. Click the boxes beside mgame.addin in the Load and Auto Load columns.

Figure 3-22: Plug-in Manager window



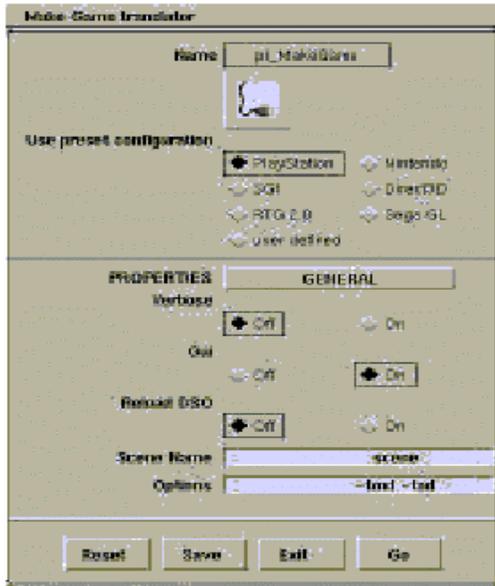
6. To open the Make Game translator window, select MakeGame from the Utilities menu.

Figure 3-23: Opening the Make Game translator window



7. Select the PlayStation radio button from the Game Box section.

Figure 3-24: Make Game translator window



This will set up the default settings to use with the PlayStation development tools translator. You can then change any of the settings for the particular type of output you require.

8. To run the translator, click the Go button at the bottom of the Make Game translator window. The PlayStation Export window is displayed.

Notes

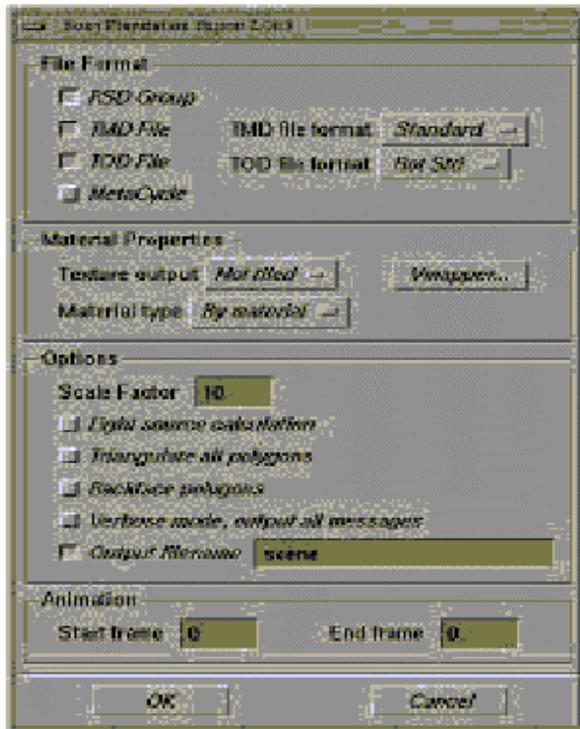
The default setting for PlayStationExport is the GUI (Graphical User Interface) mode, which gives you access to the Video Mapping (VMapper) utility for placing texture images.

While the GUI mode translator is running, the Alias window is "frozen" and will not update. You may see some ghost images from the PlayStation Export and VMapper windows until the translator is complete and control is returned to Alias. If the GUI mode is turned off, the settings from the Make Game translator window are used as options to the command line invocation of the translator, and PlayStationExport does not output texture information.

PlayStation Export window

As well as setting various options such as file formats and the animation start and end frames, this window gives you access to the Vmapper... window.

Figure 3-25: PlayStation Export window



File Format section

The PlayStation utility can output several of the PlayStation file formats including the RSD group (PLY, MAT, GRP, and RSD files), TMD, TOD, and TIM image files.

- RSD Group**
 The RSD Group option outputs the RSD group (PLY, MAT, GRP, and RSD files) for all geometry loaded in the scene.
- TMD File**
 The TMD File option outputs the TMD file for all geometry loaded in the scene. If the toggle button beside TMD File is selected, you can change the TMD file format by selecting the items in the popup menu:

Figure 3-26: TMD File option



- **Standard**
The Standard option outputs a TMD file for the selected frames of an animation. Frames are selected using the Start frame and End frame boxes at the bottom of the PlayStation Export options window.
- **Difference**
The Difference option outputs VDF and NDF file for the selected frames of an animation for MIME animation.
- **Sequence**
The Sequence option outputs a sequence of TMD files for the selected frames of an animation.
- **TOD File**
The TOD File option is only available if the scene contains animation data. Selecting this option outputs a TOD file for the animation. TOD format can be either Rotation matrices or TRS components.
- **Rotation matrices**
These are more accurate, and transfer the Alias rotation matrix directly.
- **TRS components**
This takes the Alias rotation matrix and decomposes it into TRS values that can be used in the Alias viewer program. The axis orientation is slightly different than that of the "standard" PlayStation TOD demo program, and needs to be transposed for pre/post multiply differences. Both formats have an option to suppress the output of data for objects that have Identity matrices or TRS components.
- **MetaCycle File**
The MetaCycle file option outputs a NET file, which is the animation definition file for the MetaCycle previewer. If the MetaCycle option is not selected, the selected range of animation will be output.

Material Properties section

The Material Properties section contains options for customizing your texture output attributes.

Texture output menu

These options are used to determine how the texture (TIM files) are output.

- **Off**
The Off option turns texture output OFF.
- **Tiled**
Select the Tiled option if the UV coordinates for the texture are out of the 0.0 - 1.0 range. This option tiles the texture to account for the texture wrapping. The size of the texture is maintain.
- **Not tiled**
If you choose Not tiled, no tiling occurs on a texture if the UV coordinates are out of the range 0.0 - 1.0. The UV coordinates are clipped to a 0.0 - 1.0 range.

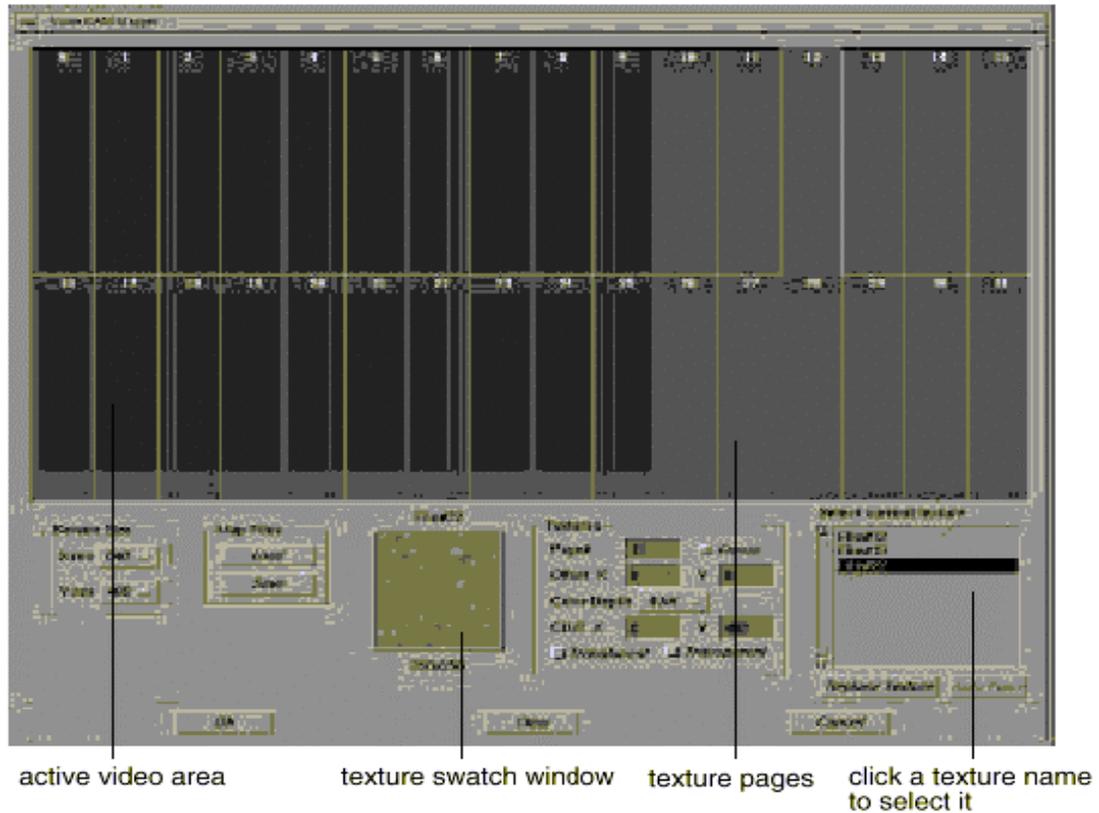
Material type menu

- **By material**
This option outputs the texture according to the materials assigned to it in PowerAnimator.
- **Flat**
This option is used to convert all materials to flat shading.
- **Gouraud**
This option is used to convert all materials to Gouraud shading.

Video RAM Mapper window

The Video RAM Mapper, also called the VMapper, allows you to place the textures used in the scene into the video RAM of the PlayStation development system. The video RAM contains 32 pages, as indicated by the two rows of vertical bars. Click the Vmapper button to display the Video RAM Mapper window.

Figure 3-27: Video RAM Mapper window



Placing textures

To place a texture into video RAM, select a texture from the Select current texture list by clicking its name. The texture you select appears in the texture swatch window. Place the texture in a page by clicking in the texture page window. The texture appears as a rectangle. The rectangle represents the size of the texture in video RAM. You can click-drag the texture to fit as many in each page as possible. Some rectangles may overlap more than one page, but you cannot place a texture over the active video area (the light blue area) of the video RAM. The texture pages with a few textures placed appear as in the following: Each texture page is numbered from 0 to 31 and represents a texture page in PlayStation development system hardware.

Active Video Area

The active video area (the light blue rectangles) is the area reserved for the active video screen. You cannot place a texture in this area. You can modify the size of your active video screen using the Screen Size menus in the lower left corner of the Video RAM Mapper window.

Screen Size Menu

The Screen Size menus specify the size of your video screen. Changing these values changes the video screen area in the texture page window.

Figure 3-28: Screen Size Menu



X res specifies the X resolution of the video screen area.

Y res specifies the Y resolution of the video screen area. The 240 option represents single buffer. The 480 option represents double buffer

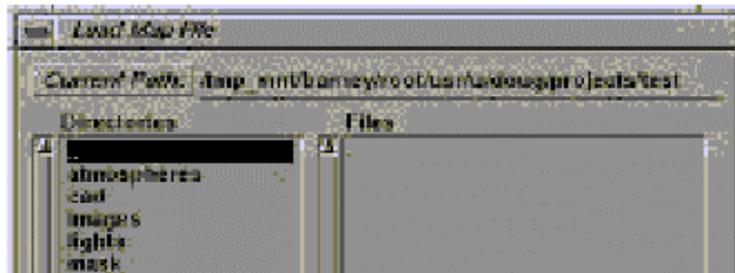
Save Map File window

If you click the Map Files Save button, the following file browser is displayed so that you can save the video RAM map (vrm) to a file in your DT_VRM_DIR directory:

Load Map File window

If you click the Map Files Load button, the following file will be d

Figure 3-29: Load Map File window



This file places textures where they were previously saved in video RAM. The texture swatch displays the selected texture.

Note: If the textures have not been loaded in the current session, they will appear black and you will not be able to move them.

Note: If the PlayStation DSO is not set to reload each time it is selected, the texture swatch area may not be updated properly on the screen. The textures will be output correctly.

Textures section

The Textures section displays the current video RAM information for the selected texture.

Figure 3-30: Textures section



- **Page #**
This represents the current texture page in which the texture resides.
- **Offset X/Y**
This value is used to specify the X and Y offset into the texture page.
- **Color Depth**
This popup menu is used to specify the color depth for the texture to be output.
 - If you choose 4 bit, the texture rectangle is the smallest possible and has a yellow outline.
 - If you choose 8 bit, the texture rectangle is medium size and has a green outline.
 - If you choose 16 bit, the texture rectangle is the largest possible and has a blue outline.
- **CLUT X Y**
This value specifies the position of the color look-up table (CLUT) in the video RAM.
Note: After changing a value in any number field, you must press either the Enter or Return key for the new value to be used.
- **Freeze**
This allows the texture to be frozen in position and not accidentally moved.
- **Translucent**
This option controls the STP bit of the color entries. See the following table.
- **Transparent**
This option controls the STP bit of the color entries. See the following table.

Table 3-3: Controlling the STP bit of the color entries

Translucent	Transparent	Black STP	Color STP
not set	not set	0	1
not set	set	1	1
set	not set	0	0
set	set	1	0

Select current texture section

The Select current texture listing lets you to select a texture. When you have placed all the textures in your animation and have customized the settings the way you want them, click OK to export the file.

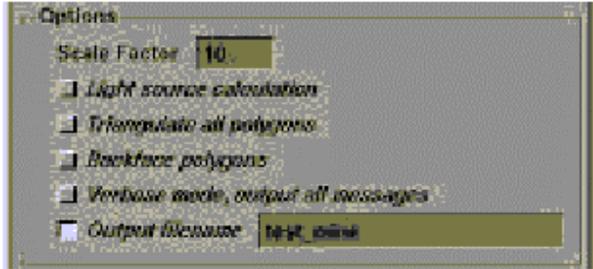
If you decide not to use the textures showing in the window, click Clear.

If you want to close the window without exporting the file, click Cancel.

The name shown here is the name of the texture as set in the multi-lister shader texture window.

Options Section

Figure 3-31: Options Section



Type in the Scale Factor box to scale the model on output by a specified factor.

Toggles

- If Light source calculation is toggled ON and there is a light in the scene, the light is replaced with a material that has been calculated to simulate a light source.
- If Triangulate all polygons is toggled ON, all polygons in the scene are triangulated.
- If Backface polygons is toggled ON, backfacing is turned on in the MAT file.
- If Verbose mode, output all messages is toggled ON, all messages are output to STDERR during the export.
- If Output filename is toggled ON, you can select the output filename for the TMD and TOD files.

Animation Section

Figure 3-32: Animation Section



This section lets you select an animation range for the TOD file format output.

- Enter a number in the Start frame box to set the beginning frame of the animation.
- Enter a number in the End frame box to set the last frame of the animation. When you have set all the options in the PlayStation Export options window the way you want them, click the OK button to export the file.

GetPlayStation Command Line Options

The GetPlayStation Dt Translator has the following options that can be used with the mgame plugin with the option field.

GetPlayStation Options

Table 3-4: GetPlayStation Options

-v	Verbose mode, print out all messages
-rsd	Output RSD group
-tmd [option]	Output TMD file: normal difference sequence
-tod [option]	Output TOD file: 0 (r) Rotation matrices not optimized 1 (R) Rotation matrices optimized 2 (t) TRS not optimized 3 (T) TRS optimized
-net	Output Character Network file.
-flat	Convert all material to flat shading.
-gouraud	Convert all material to gouraud shading.
-s #	Set scale of models
-lightSource	Turn Pre-light sourcing ON
-b	Turn backface polygons ON
-t	Triangulate all polygons
-x scriptname	Shell script to run at completion
-o [filename]	Set output filename of TOD file
-d	Output Dynamic Properties file

Most of the command line options match the GUI user controls. See the description of the PlayStation Export window on page 5 for details of the settings.

Options with no corresponding GUI mode

-x Lets you specify a shell script to run at the end of the translation process. The PlayStation also looks at the following environment variables:

DT_PSX_BEGSCRIPT

DT_PSX_ENDSCRIPT

for commands that should be run at the beginning and the end of the translation process. The end script process is given the name of the output scene directory and name as command line options.

Additional user controls with environment settings

- **DT_PCNames**
This changes the names of the output files to be in a PC dos compatible format of DTF#dddd where dddd is a 4 character digit. At the moment, there is no easy mapping of the numbers to the original files.

Note: File names that fall within the DOS limitation of 8.3 are not changed.
- **WF_TIM_DIR / DT_TIM_DIR**
Use this command line option to pre-define the directory to open when using the replace texture button.
- **DT_VRM_DIR**
Use this command to specify where to find the vrm files that can be saved and/or loaded.
- **DT_EXPORT_DIR**
Use this command to specify where the exported files should be placed. They will be displayed in the Export directory field of the mgame plug-in.
- **PSX_RSD_SCALE**
Use this command to use the 7.5.1 method of saving UV coordinates in the RSD material file. The default is to use the texture UV size and not the position of the texture in the VRAM space.

Note: Environment variables can be entered into the text fields of mgame and will be expanded internally.

PlayStation Demo Viewer Setup

The following information shows how to set up and use the example viewer program for the PlayStation development system.

Demo Viewer

The demo viewer for the PlayStation development tool is based on one of the demo TOD viewers from the PlayStation programmer tools CD. The program has been modified to support the MetaCycle feature of Alias and the PlayStation translator. Also, due to differences in the pre/post multiply differences in matrix applications the component (TRS) packet code for the viewer has been tuned to the output of the translator. The rotational matrices packet is the same for the original demo program. It views TMD files and TMD/TOD files, with or without texture TIM files. It also supports the MetaCycle NET files that are created with the PlayStation translator GetPlayStation.

The files that are found in this area are used for running the PlayStation utilities on an Intel platform. (These were used under Windows 95).

chk_bios.com	executable to check for existence of dexbios and return errorlevel depending whether or not it is loaded.
makefile.mak	make file for building the example viewer netv.c
netv.bat	batch file to load in the TMD/TIM/TOD/NET files to the expected memory locations for netv.cpe

The batch file looks for the files test.tmd, test.tod, test.net, and *.tim files in the user supplied directory.

Run as:

```
netv h:\GET\download
```

where

```
h:\GET\download
```

will have the files test.tmd...

The batch file is setup to look for known file names (test.*)in the directory specified as a parameter.

Unix Scripts

The unix scripts psx_begscript and psx_endscript have been supplied to show how one might use them to implement a one button download from Alias to the PlayStation development system.

On the SGI workstation, you have to set the environment variables PSX_BEGSCRIPT and PSX_ENDSCRIPT to the full path names of shell scripts to be run before and after the translation process takes place.

netv.c	source code to the example viewer
netv.cpe	example viewer executable file
rerun.bat	batch file to check for dexbios loaded and reset the PSX development boards if necessary.
shownetw.bat	batch file to check for the file "test.use" in a user supplied directory, to see if there is a new file to be loaded into the PSX development system.

This can be used as a simple auto-load script for continuous hands-free one button downloading from within Alias. You would normally have to have the PC on the network with the SGI workstations. There are many ways to do this, one of which is using Samba (a freeware program from Australia that runs on the SGI and makes it look like a Microsoft network machine).

Installation Notes

Installing PlayStation Translator

This section describes how to install a translator as supplied directly from Sony Computer Entertainment's BBS or developer's CDROM.

Distribution of PlayStation Translator

The PlayStation translator is distributed by Sony Computer Entertainment on their BBS and developers CD. The file that can be downloaded or copied from the CD is an un-encrypted compressed tar file containing the PlayStation translator, PlayStation TIM image module, source code and make file to build the translator locally.

To install the PlayStation translator, use the InstallGame script file that is found in the \$ALIAS_LOCATION/ODS/Games/bin directory. This script is installed as part of the standard Alias installation process.

This script file takes the compressed tar file, uncompresses it, and untars it to the location you specify. There are command line arguments to override the default installation locations.

Installing the new downloaded version of PlayStation Translator

The shell script InstallGame is run to extract the components of the translator and place them into the proper directory tree for use within the Alias game tools environment. The command line options that are available for InstallGame can be found by running the script with the parameter -h as shown below:

```
$ALIAS_LOCATION/ODS/Games/bin/InstallGame -h
```

outputs the command line options that can be used.

Usage: InstallGame <-hdeafsk>

where: -h prints this usage info

-e assume files are not encrypted

-d decrypt the library file only

-a dir alias directory location

-f dir install from this directory

-t dir install to this directory

-s System install this System

Default with no options on command line is to use the \$ALIAS_LOCATION/ODS/ Games/support directory as the source directory and to install back into the ODS/ Games directory. Use absolute paths for the from and to directories

Note: When running as root, ensure that \$ALIAS_LOCATION is set correctly

What you need to know:

1. Where the original compressed file is located
2. Where the expanded file is to be placed. (\$ALIAS_LOCATION/ODS/Games)
3. Where Alias 8.0 is installed

Defaults:

1. Original compressed file has no default location. It is wherever you placed it.
2. For an expanded file, the default is in the Games area under Alias 8.0 installation
\$ALIAS_LOCATION/ODS/Games
3. Alias 8.0 defaults to /usr/aw/alias if ALIAS_LOCATION is not set.

Access:

If the installation is to write out to the Alias installation area, you will need to have access to the root password in order to become the super-user. For local installations, root access is not necessary. For example, to change to root and install the PlayStation translator from the compressed tar file SO in your home directory to the expanded tree under the Games tree of Alias 8.0, type:

```
su -l root
```

```
setenv ALIAS_LOCATION /usr/aw/alias
```

```
ALIAS_LOCATION/ODS/Games/bin/InstallGame -e -f $HOME -s SO
```

How to run the translator

The translator is run from within Alias using the Makegame plugin. See the Games Data Transfer book for more information on makegame usage.

How to rebuild the translator

The following method is recommended to rebuild the PlayStation translator:

1. Create a games tree under your \$HOME directory somewhere.

```
Fill it with the following directories: lib, include, Dtplugins, PlayStation
cd $HOME
mkdir Games
cd Games
mkdir lib
mkdir include
mkdir Dtplugins
mkdir PlayStation
```

- Copy the source files from the following directory

```
$ALIAS_LOCATION/ODS/Games/source/PlayStation/src
```

to this new Games/PlayStation directory

```
cp $ALIAS_LOCATION/ODS/Games/source/PlayStation/src/* PlayStation
```

- Check the Makefile for the proper setting of GAMES_LOCATION.

For example:

\$SITEROOT defines the "Site" directory of the development tree. It is a relative path from this current directory. All other \$variables in this Makefile should be relative to SITEROOT.

```
$SITEROOT = $(GAMES_LOCATION) INCDIR = $(SITEROOT)/include
```

GAMES_LOCATION set to \$ALIAS_LOCATION/ODS/Games will then use standard Games include and library files. If the library DtOMio is to be changed as well, it is easiest to point GAMES_LOCATION to the root of the "local Games" tree. (In the example above, \$HOME/Games.) Make sure that the files in the include directory are copied over to the modified Games tree.

- Type make

This should rebuild the translator DSO GetPlayStation

- In order for makegame to see the new version of the DSO, you will need to set the following environment variable.

```
setenv DT_PLUGIN_LOCATION $HOME/Games/PlayStation
```

Using standard libraries, you only need to set DT_PLUGIN_LOCATION to point to the location where the DSO is found.

Note: The PlayStation translator is normally located in the same directory as GetPlayStation

Using modified libraries, set GAMES_LOCATION to point to the root of the modified Games tree.

Note: The libDtxxx.so files from the \$ALIAS_LOCATION/ODS/Games/lib directory should either be copied to this new tree or have symlinks created.

- Run Alias or DtExport to test the new version of the translator.

Release Notes

Version 2.0b3 Improvements

- control of STP bit in TIM files has been added
- control of material UV coordinates has been extended

Version 1.0 Improvements

- 256 and 512 PAL screen sizes have been added
- The default TIM depth has been changed to 4 bit

Limitations

- Solid textures are not supported directly in the Dt translators
- TOD format for TRS data is in Alias orientation and not standard TOD.
- The example TOD viewer supplied with the Alias translator shows the matrix order that the angles are supplied in.

Modeling Limitation

TOD format has a maximum scaling factor of 8.0 in the 16 bit fixed point format that is specified. Scaling is (1,3,12), with 1.0 = 4096.

Model in Alias so as the scaling components of the transformations are less than 8.0.

User Interface - WorkFlow notes

- When entering numeric data into the fields on the VMapper window, you must press Enter to actually set the data
- Similar to the above, when you change focus, the values in the numeric text fields will only be set when you press the Enter key.
- Screen area definition does not relate to programming (for example double buffering using the 2x screen area)
- Running the VMapper window multiple times causes the texture swatch area to become corrupted. Textures still output correctly. (Run the translator in "Reload DSO" mode if this is a problem, but note that it will start up slower.)
- It is possible to position TIM over a page boundary illegally when the texture is placed between two other textures already placed.
- It is possible to overlap a larger TIM over a smaller TIM.
- Default numbering of color look-up tables (CLUTs) may position them over the 511 limit.

TOD_I.KXP: Plug-in for 3D Studio

Version 4.0f

Documentation date: 12/26/95

Document

3dstod_e.txt (English)

3dstod_j.txt (Japanese)

Installation and behavior check

Installation method is written in the document. In order to check the "Previewer" feature of the tool, load sample data containing animation data into 3D Studio, and invoke the plug-in. Then click the Preview button. If the tool is running correctly, you will see the animation running on DTL-H2000.

In order to check the "Save RSD" and "Save TOD" features, click "Save RSD" button, and "Save TOD" button. Check the data by giving those data to the TOD previewer on DTL-H2000.

Sony Computer Entertainment makes no expressed or implied warranty of any kind with regard to these programs or to the supplemental documentation in this manual. In no even shall Sony Computer Entertainment be liable for incidental or consequential damages in connection with or arising out of this program or documentation. This disclaimer includes, but is not limited to, any loss of service, loss of business, or anticipatory profits, or consequential damages resulting from the use or operation of the enclosed software.

3D Studio is a trade mark of Autodesk Inc.

PlayStation is a trade mark of Sony Computer Entertainment Inc.

Function

This program is a plug-in program for the Autodesk 3D Studio (3DS).

With this plug-in, you can save 3DS object or animation data in PlayStation format (RST ,TOD) while using 3DS. You can also preview animation while using 3DS by sending animation data to PlayStation board (DTL-H2000).

Release History

Version 4.0f

Bug Fixes

1. Texture map UV value had been incorrect.

Version 4.0e

Bug Fixes

1. Dummy object's parent was not correctly saved into TOD file.
2. Texture UV was incorrect in some cases.

For instance, if a texture pixel size is $w * h$,

the correct UV range is (0,0) to (h-1, w-1),

however, the UV range was shown as (0,0) to (h,w).

3. TOD preview is now compiled with latest (version 3.*) library.

You have to re-install the previewer. Please read installation section.

Version 4.0d**New Features**

1. You can save hierarchy data into TOD file.
2. You can select RST (Rotation, Scale, Translation) packet type.
3. Load object from TOD plug-in dialog.
4. Save (and load) plug TOD plug-in parameters into TOD.INI file.

Bug Fixes

1. There was a bug in scaling objects.

Version 3.1

1. Object scale is converted to TOD data.
2. Some sample data (for instance, IK_*.3DS) could not be handled in version 2.1. This version can handle them.
3. You can preview or save animation without re-invoking the plug-in.
4. You can specify start and end frame of an animation to preview or save.
5. Objects with more than 10 textures could not be processed in version 2.1. This version can handle them.
6. In "Quick Preview" only TOD file is generated. TMD files are generated only once when you invoke "Preview".
7. You can save RSD file and TOD file independently.

Version 2.1

1. You can save camera animation data into TOD, and preview it.
2. You can save mesh objects in RSD files.
3. You can configure the plug-in environment.

Requirements

This plug-in runs only on 3DS R4.

In order to use the previewer function, you also need DTL-H2000.

Files

This plug-in kit contains the following files:

3dstod_e.txt	Document in English
3dstod_j.txt	Document in Japanese
tod_i.kxp	Plug-in module.
tod.ini	Configuration file sample.
psboard.ini	DTL-H2000 port address configuration file.
psbload.exe	Communication program between DTL-H2000 and PC
psreset.exe	Communication program between DTL-H2000 and PC
psrun.exe	Communication program between DTL-H2000 and PC
patchj.exe	DTL-H2000 patch program (Japanese version)
patchw.exe	DTL-H2000 patch program (non-Japanese version)
todvj.bat	TOD previewer batch file (Japanese version)
todvw.bat	TOD previewer batch file (non-Japanese version)

todview5.exe TOD previewer
 todview5.c TOD previewer source code

If you use do not use the preview feature, you need only three files, "tod_i.kxp", "tod.txt", and "tod.ini".

Installation

1. Copy plug-in module "tod_i.kxp" to the "process" directory in the 3DS home directory.
 If you have installed your 3DS into "c:\3ds4", then you have to copy "tod_i.kxp" to "c:\3ds4\process" directory.
2. Copy configuration file "tod.ini" into 3DS home directory.
 If you do not use the TOD previewer, installation is complete.
 If you do use TOD previewer, follow the installation steps below.
3. Copy "psboard.ini" to root directory "c:\".
 This file specifies the port address of DTL-H2000.
 If your DTL-H2000 card's port address is 0x1340, then the file must contain a line as follows.
 addr=0x1340
4. Copy all the following files to "c:\psxgraph\bin".
 todvj.bat
 todvw.bat
 psbload.exe
 psreset.exe
 psrun.exe
 patchj.exe
 patchw.exe
 todview5.exe

 If you are using the Japanese version of DTL-H2000, you have to rename "todvj.bat" to "todv.bat". If you are using the non-Japanese version of DTL-H2000, you have to rename "todvw.bat" to "todv.bat". You can change the location of these programs. Please read the customization section.
5. Create working directory "c:\tmp" for the previewer. You can change the working directory. Please read the customization section. Installation is complete.

Notes

The TOD previewer does not work properly if "DEXBIOS" is running on your PC. If you have one, you can delete them from PC memory as follows:

```
C:\> dexbios
```

If you see a message "Removing DEXBIOS" on the screen, then "DEXBIOS" is removed from memory.

Be aware that if you invoke the command when "DEXBIOS" is not running, this causes DEXBIOS to start.

You will then need to invoke the dexbios command again until you see the "Removing DEXBIOS" message on your screen.

How to start

1. In the KeyFramer of 3DS, press "F12" key.
2. Select "TOD" from KXP plug-in list.

Previewer key bind

The previewer's key bind is as follows:

L1, L2	Change BG color
SELECT	Pause
START	Reset Pause

You can proceed frame by frame, using SELECT key while you are in the Pause mode.

Functions

Scale

Scales size of object size and translation of object/camera.

For instance, if you set scale value to 10.0, then the object size and translation of object and camera is expanded 10 times.

You can specify the scale value to control the object size. Also, you use this value to get rid of the near-clip, far-clip and z-sort problems, which will occur on PS.

Frames from ___ to ___

Specify regions of animation frames you want to preview or save. You can use this function to extract some portion of animation, or to divide one animation to some pieces.

TOD type

Specify TOD packet type.

Matrix	MATRIX (Rotation Matrix + Translation)
RST	RST (Rotation vector, Scale, Translation)

Animation will look same whether you chose Matrix or RST. RST data is converted to Matrix on PlayStation, therefore RST type TOD animation will consume much CPU power.

Hierarchy

On	Hierarchy relation of objects is saved.
Off	Hierarchy relation of objects is not saved.

The animation will look same whether you chose On or Off. When you chose On, each object's animation data will be relative to its parent object. Whereas if you chose Off, the animation data will be relative to the WORLD.

You do not have to save hierarchy information as far as you play the TOD data. But, if you have to animate a body of a robot with you program, for example, hands and legs of the robot will be disconnected because they are not linked to the body. In such a case you should save hierarchy information in the TOD file.

In order to support hierarchy structure, dummy objects in 3DS will be saved in TOD file. Dummy object will be represented as an object which has coordination data only (TMD is null).

Frames

Thin down the frame data.

All	saves all frames
1/2	saved every 2 frames
1/3	saved every 3 frames
1/4	saved every 4 frames
1/5	saved every 5 frames

If you specify "1/4", then only frames "1,5,9,13,..." will be saved. This will make the TOD data size 1/4. You can interpolate between frames to reproduce the original animation sequence on PlayStation. You can use this method to reduce the size of TOD data.

The TOD previewer included in this tool, does not do interpolation. If the previewer finds some frames missing, then it will draw the previous frames again. For instance, if you had specified "1/3", then the frames will be like this.

TOD frames: 1, 4, 7, 10, ...
 Drawn frames: 1,1,1,4,4,4,7,7,7,10, ...

Load obj

Loads new 3DS file without exiting from TOD plug-in.

Preview

Invokes animation previewer on DTL-H2000.

When you click this button, object data is saved as temporal RSD files. Then RSDLINK command converts the RSD files into one TMD file. Then, animation data is save as one TOD file. The TMD file and TOD file are transferred to DTL-H2000, where TOD previewer is invoked.

You can speed up the invocation of the previewer by placing the working directory (specified as TMP_DIR in TOD.INI file) in the RAMDRIVE of you PC.

Quick Preview

Invokes animation previewer on DTL-H2000.

Only TOD file (animation data file) is generated. Instead of generating new TMD file (object data), previously generated TMD file found in (c:\tmp\tmp.tmd) is used for previewing.

Quick Preview is useful when you have finished editing objects and you are frequently changing animation data only. In such cases, you invoke "Previewer" only once, and you can invoke repeatedly "Quick Previewer" after that. This speeds up the previewing process, because "Quick Previewer" skips generating RSD files and TMD file.

In order to make object scale and animation scale reasonable, you have to set the scale value of "Preview" phase and the "Quick Preview" phase same. If you need to change the scale value, you have to invoke "Previewer" in order to save TMD file (object data) in the new scale.

Save RSD

Object data is saved as RSD file.

Each object is saved as different RSD file.

System will ask "Generate File name automatically?".

If you choose "Yes", system asks you a base name which will be used to generate new file names. System adds hexadecimal 3 digit to the end of the base name. Therefore only first five letters are used as base name.

For instance, if you give "c:\psxgraph\data\rsd\mydata.rsd" as the base name, following RSD files will be generated automatically.

```
c:\psxgraph\data\rsd\mydat000.rsd
                               mydat001.rsd
                               mydat002.rsd
                               ...
                               mydat00e.rsd
                               mydat00f.rsd
                               mydat010.rsd
                               ...
                               mydatfff.rsd
```

The object names used in 3DS cannot be used as RSD files, because it can be longer than 8 letters.

TOD plug-in also generates PRJ file (in the case above, "mydata.prj"). Correspondence between object name and RSD file is specified in the project file.

If you choose "No", you must give all the RSD file names.

Select "Yes", except when you have to specify the file names by yourself.

Material

3DS material should be set as follows in order to generate Gouraud polygon in RSD file.

1. Polygon is either, Gouraud, Phong, or meta.
2. At least one of its edges is smooth. which means that, it has at least same smoothing group with at least one of its adjacent polygons.

"Diffuse color" in 3DS is converted into polygon color in RSD.

"Non 0 transparency value" in 3DS is converted into semi-transparent polygon in RSD.

Texture maps

1. Only the "texture 1" is converted to RSD.
2. Texture more than 256x256 can not be correctly converted to RSD.
3. Tiling information is ignored.
4. Face map can be converted to RSD.

Any other 3DS material information is ignored.

[?]

Each object is saved as RSD format in local coordinate. The difference of each object from the origin of world coordinate is saved in TOD file. For instance, if you have created a human body which consists of a number of parts, and saved each part as a different RSD file, those RSD file objects do not represent the location of each part in the body. In order to take account in the location of each object, you can follow the following method,

1. On PlayStation, specify each object's coordinate referring the TOD data.
2. On 3DS attach all the parts in one object.

TOD previewer is using the first method. Second method is valid only when each part of the body does not change the location of each other.

Save TOD

Keyframe animation data is saved as TOD file.

When you have many animation data for one character, you only have to save object data (RSD file) only once, and save each animation as a different TOD file.

Limitations

1. Morphing data is not converted.
2. Information on lights is not converted.
3. Pivot:

3DS's pivot is interpreted as translation and transformation in TOD data. Therefore as far as you play the TOD data on PS, the animation on 3DS is correctly reproduced on PS. But, the data of "pivot" itself is not converted to TOD data. So you cannot animate an object according to its pivot from your program. For instance, if you place a pivot of a robot's arm to its shoulder point, and if you try to rotate the arm around the shoulder point, the arm will rotate around its local axis, not around the shoulder point.

You can use "dummy object" instead of pivot, to change the rotation axis.

4. Link information:

In 3DS, you can specify whether to propagate the rotation and scaling information from the parent object to the child object or not. On the other hand, PlayStation's library (libgs) always propagate those information from the parent to the child. Therefore, if you specify not to propagate any of those information on 3DS, the resultant TOD will not reproduce the same effect. To avoid those problems, you should not stop propagating the information from the parent to the child.

5. Scaling:

If the scale value exceed 8.0 on 3DS, it will not apply properly on PlayStation. It is because, PlayStation uses fix-point for the scale value. "8.0" is the maximum scaling value expressed in fix-point (4096 * 8). Notice that, in hierarchly linked object, scale values are multiplied from all thorough it ancestors. So the scale value could exceed 8.0 even if its local scale is less than 8.0.

Generated Files

TOD plug-in generates following files.

1. Preview

In temporary directory (c:\tmp),

RSD files
tmp.3ds
tmp.prj
tmp.tmd
tmp.tod

2. Quick Preview

In temporary directory (c:\tmp),

tmp.tod

3. Save RSD

In temporary directory (c:\tmp),

tmp.3ds

In user specified directory.

RSD files
PRJ file

4. Save TOD

In user specified directory.

TOD file

Customization

The TOD plug-in invokes TOD previewer as follows.

```
c:\psxgraph\bin\todv c:\tmp\__000__.tmd c:\tmp\__000__.tod
```

The argument specifies an TOD file to preview. This file is created by the TOD plug-in. You can substitute the previewer batch command "todv.bat" with your own previewer.

Information

If you have any question on this program, please contact us (SCE).

It is useless to call Autodesk, because they do not take any charge of this program.

Plug-ins for LightWave 3D 4.0

Version Beta.2 (Intel, Alpha, SGI)

Documentation date: 5/1/96

Installation

1. Copy the appropriate plug-in file for your system to your plugins/modeler and plugins/layout directory.

To plugins/modeler:

RSDLOAD.P (RSD Geometry file loader)

RSDSAVE.p (RSD Geometry file saver)

To plugins/layout:

TIMIMAGE.P (TIM Image file loader and saver)

2. Add RSDLOAD.P and RSDSAVE.P to your Modeler

Start LightWave Modeler 4.0 and choose "Add Plugin" from the "Objects/Custom" pop-up.

Select the plug-ins from the file requestor. In the "Custom" menu, you will find "RSDLOAD.P" and "RSDSAVE.P" added to the list. When you quit Modeler, the plug-ins will be saved in the list for future use.

3. Add TIMIMAGE.P to your LightWave

Start LightWave 4.0 and choose "Add Plug-ins" from the "Options" pop-up.

Select the plug-ins from the file requestor. In the "RGB Image Format" popup from the "Record" menu, you will find "TIMIMAGE.P" added to the list. When you quit LightWave, the plug-ins will be saved in the list for future use.

Instructions

RSD file is the geometry file for the PlayStation. It has 4 separate files such as (.rsd), (.ply), (.grp) and (.ply) file.

RSDSAVE.P exports above files from Modeler. To export (.rsd) files, select "RSD_EXPORT(Ver.1.0)" from the "Custom" popup in the "Object" menu and type a file name to export your geometry without suffix from the file requestor.

Tips: PlayStation can read only 3 and 4 vertices polygons. If the geometry has >4 vertices polygons, then the RSDSAVE Plugin show you the error message dialog. You can triple >4 vertices polygons with T-key after finding >4 vertices polygons with the "Stats" button from the "Display" menu (same as w-key).

RSDLOAD.P imports the (.rsd) files to Modeler. To import (.rsd) files, push the "Load" button from the "Objects" menu. And choose the (.rsd) file to import from the file requestor. the RSDLOAD automatically reads the (.rsd) file included (.ply) and (.grp) files to your Modeler.

Note: PlayStation and LightWave have different texture mapping method. You lost all of texture vertices in the (.rsd) file in the Modeler when you load your (.rsd) file with texture uv vertices in the (.mat) file.

TIMIMAGE.P load and save Sony TIM image format files to the LightWave. You can use any TIM image file as the texture mapping images in your animations and you can save your rendered image as the TIM image files.

TIMIMAGE.P supports following TIM image format.

To load: 4bit CLUT (Color Look Up Table)
 8bit CLUT
 16bit Direct color
 24bit Direct color

To save: 24bit Direct color

To load the TIM image files to your LightWave, select "Load Image" button from the "Images" menu, and choose the (.tim) file to load from the file requestor.

To save the TIM image files from your LightWave, select "PlayStation(.tim)" popup from the "RGB image format" in the "Record" menu, and specify (.tim) file name from "Save RGB images" and render your scene.

Caution: Do not copy these plug-ins to anybody. They are still beta version.

Chapter 4:

Sound Conversion Utilities

SMF2SEQ: Creates Sequence Data (Mac version)

SMF2SEQ converts a format 1 standard MIDI file (SMF) created by such sequence software (score editing software) as Performer™ (Mark of the Unicorn) and Vision™ (Opcode) into the sequence data format (SEQ file) for the PlayStation.

Table 4-1: Appropriate Format (SMF)

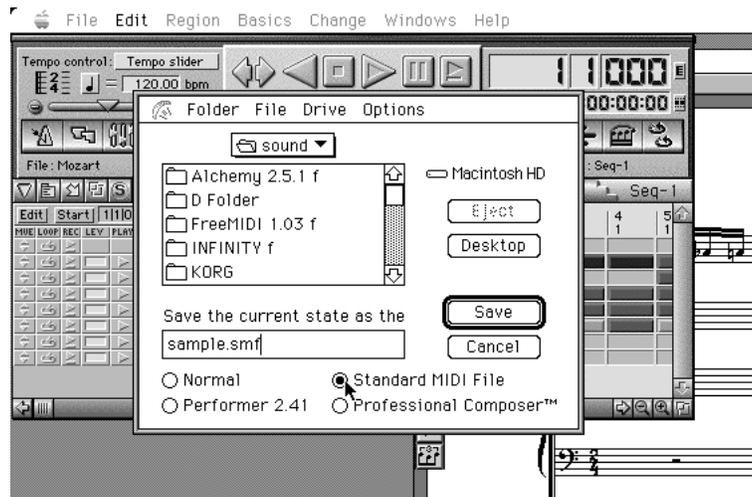
Appropriate format	SMF (Format 1)
Sampling frequency	Optional
Appropriate message	Note on/off Program change Pitch bend Control change

Resolution in SEQ is optional. But the highest resolution depends on the actual reproduction mode (1/60 sec., 1/120 sec., 1/240 sec., etc.).

Setting Up SMF File

Before using SMF2SEQ, save the sequence to be converted by your sequence software in the SMF format (format 1).

Figure 4-1: SMF Output in Performer

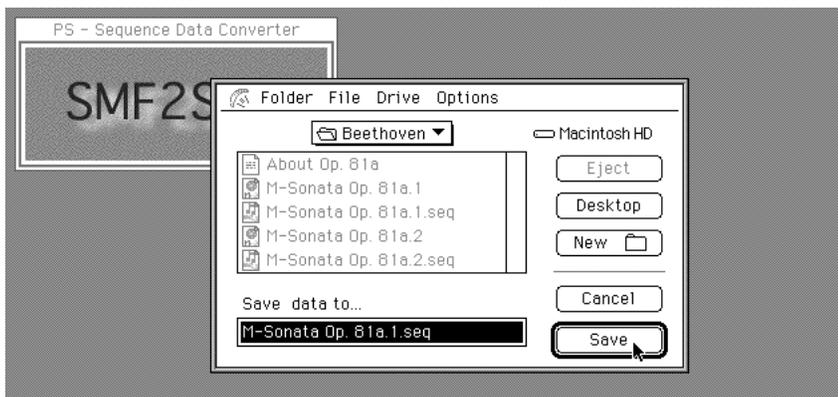


SMF2SEQ - Creating Sequence Data

Using SMF2SEQ

1. Double click the SMF2SEQ icon to start up the application.
2. Select “Convert” in the “File” menu, and specify the file to be converted.
3. When prompted to enter the output file name, enter the file name to be assigned after conversion.
4. When conversion is complete, the MIDI message summary which was deleted during conversion will be displayed.
5. Select “Quit” from the “File” menu to terminate the application.

Figure 4-2: SMF2SEQ



An SMF file can also be converted by dragging it to the SMF2SEQ icon. (Two or more SMF files can be selected.)

Also, setting the Automatic Conversion Mode (Auto), Interactive Conversion Mode (Interactive) and BankChange Forced Deletion Mode (Remove BankChange) is possible using the mode menu.

Appropriate format	SMF (Format 1)
Sampling frequency	Optional
Appropriate message	Note On/Off Program Change Pitch Bend Control Change

Figure 4-3: SMF2SEQ Icon



SMF2SEQ: Creates Sequence Data (PC version)

Version 2.4

Documentation date: 10/26/98

Description

smf2seq converts format 1 standard MIDI file (SMF) data to a PlayStation sequence data file.

How to use

smf2seq[options]SMF-files...

Inputting a format 1 file as an argument creates an SEQ file (*.SEQ), which is a PlayStation sequence data file. Multiple files can be given to perform batch conversions. The file extension [.smf] can be used.

Options

- Q Converts using quiet mode. No warning messages are displayed.
- V Converts using verbose mode. The control change and meta events used by SMF are listed.
- B Forcibly deletes the bank change used by SMF.

Constraints

The current version has the following constraints:

- It is not compatible with SMF format 0.
- It cannot handle files larger than 64 Kbytes.
- The compatible control changes are as follows:

Bank Select (#0)

Data Entry (#6)

Main Volume (7)

Panpot(10)

Expression (11)

NRPN (98,99)

Changes in version 2.4

- A bug which caused an error code to be returned even when conversion was terminated correctly, has been fixed.

AIFF2VAG: Creates Waveform Data (Mac version)

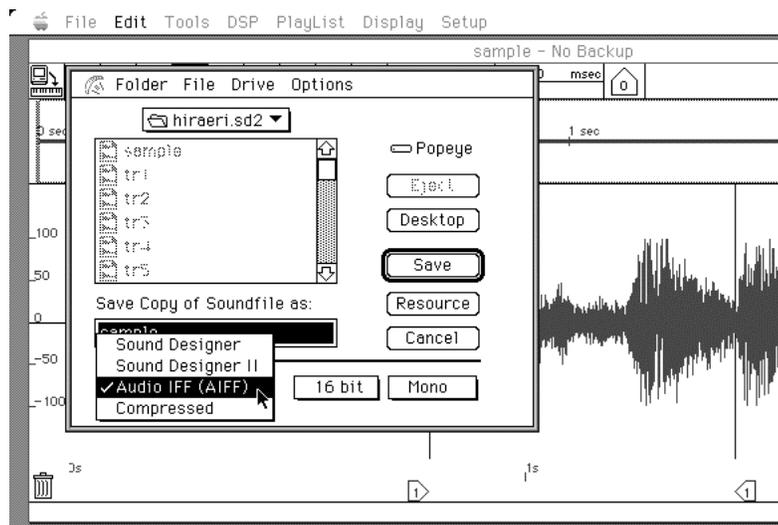
AIFF2VAG converts an Apple AIFF (Audio Interchange File Format) file created by such existing waveform editing software as SoundDesigner II™ (Digidesign) and Alchemy™ (Passport Designs) into the waveform data format VAG file for the PlayStation unit.

For VAG, a multiple of 28 of samples are handled as one block because of hardware characteristics. Thus, handling a sound source with a loop requires that the loop length be a multiple of 28.

Setting Up AIFF File

Before using AIFF2VAG, save the waveform data to be converted by your waveform editing software in the AIFF format (16 bits, monophonic, non-compressed).

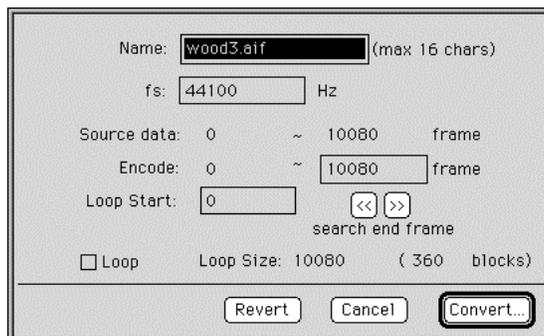
Figure 4-4: AIFF Output in SoundDesignerII



Using AIFF2VAG

1. Double click the AIFF2VAG icon to start up the application.
2. The AIFF file to be converted is requested when “Convert” is selected from the “File” menu, so specify the file at this time.
3. Specify the number of samples to be converted and the loop start point.

Figure 4-5: AIFF2VAG Conversion File



4. When the output file name is requested, enter the file name to be assigned after conversion.
5. Select “Quit” from the “File” menu to terminate the application.

An AIFF file can also be converted by dragging it to the AIFF2VAG icon. (Two or more AIFF files can be selected.)

Setting of the standard, high band components, low band components and 4bit straight compression can be carried out by means of the Compression mode menu. When noise components are conspicuous in normal compression, they can sometimes be made less so when converted in another mode.

Setting of the automatic conversion mode (Auto) or Interactive conversion mode (Interactive) and also setting of the modes related to sound reproduction after conversion (Silent, Play after converted, Show panel after conversion) is possible by means of the Conversion mode.

It is also possible to reproduce a VAG file which has already been converted by selecting "Select VAG..." from the file menu. By pressing the mouse button on the AIFF2VAG panel after selecting VAG, sound production can be repeatedly checked.

Figure 4-6: AIFF2VAG Icon



AIFF2VAG: Creates Waveform Data (PC version)

Version 3.2

Documentation date: 10/26/98

Description

The Audio Interchange File Format (hereafter AIFF) converts Windows WAV (hereafter WAV) format data and 16 bit straight PCM data (no header) into PlayStation waveform data files.

How to use

aiff2vag[options] aiff-files...

Inputting AIFF format, WAV format or 16bit straight PCM format files as arguments creates a VAG file (*.VAG) which is a PS compression waveform data file. Multiple files can be given to perform batch conversions. The file extension [.aif] can be used.

Options

- 1 Forcibly encodes waveform data with loops as a no-loop sound source during encoding.
- L Forcibly encodes waveform data with no loops as a sound source with loops during encoding.
- R fs Specifies the data sampling rate when inputting 16 bit straight PCM data. fs is Hz units.
- E Does not convert Endian (byte string sequence)

Constraints

The current version has the following constraints:

- It can only handle 16bit uncompressed data. 4bit and 8bit compressed format data cannot be handled.
- It can only handle monaural data. When converting a stereo source, divide the channels before performing the conversion.

Changes in version 3.1

- A bug in which the waveform completion flag was not correctly inserted in source data which was a multiple of 28 sample, has been fixed.

Changes in version 3.2

- A bug in which noise occurred when converting sound data with loop settings, has been fixed.
- A bug which caused an error code to be returned even when the conversion terminated correctly, has been fixed.

MKVAB: Makes VAB

Version 2.0

Documentation date: May 1, 1996

Description

MKVAB constructs a sound source bank format VAB based on the VAG format wave pattern data group created by aiff2vag.exe and the define file which creates the attribute table.

How to use

mkvab [option] *vag-files...*

Options

-f *def_file* Specifies the define file *def_file* that creates the attribute table.
 -r *vab_file* Analyzes the VAB file and outputs an attribute definition file.
 -o *out_file* Specifies the output file.

Limitations

The current version has the following limitation:

As the VAB file size and size of each VAG are automatically calculated, the specified numerical value is ignored.

Attribute Define File

VabHdr (Attribute definition covering entire VAB)

Table 4-2: VabHdr

Label	Value	Explanation
form	'VABp'	Format descriptor
ver	7	Format version number
id	0	VAB id. (Always 0)
fsize	0	File size. (mkvab is automatically calculated)
ps	0-128	Number of all programs included in VAB
ts	0-2048	Number of all tones included in VAB
vs	0-254	Number of all VAG included in VAB

ProgAtr (Program (same as instrument) level attribute definition)

Table 4-3: ProgAtr

Label	Value	Explanation
tones	4	Number of tones included in program
mvol	0-127	Program volume value
mpan	0-127	Program panning value

VagAtr (Tone level attribute definition)

Table 4-4: VagAtr

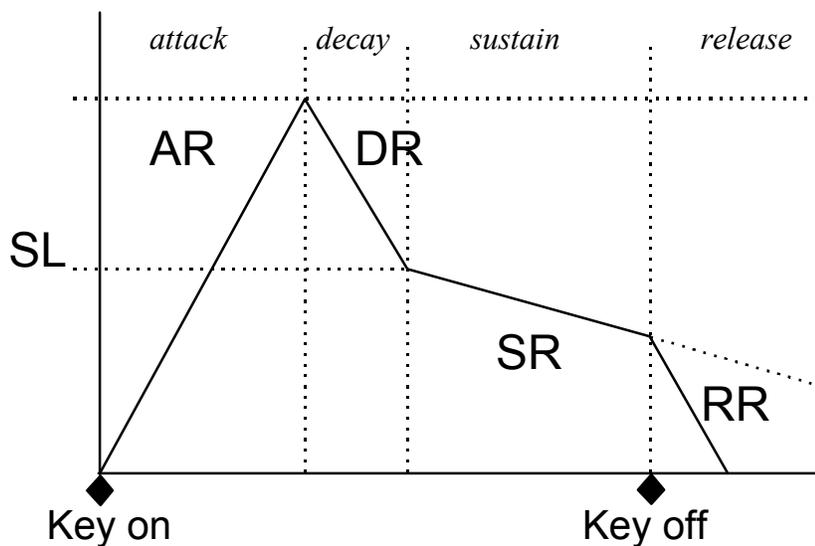
Label	Value	Explanation
prior	0-127	Tone priority level. Larger value takes precedence.
mode	0,4	When set to 4, reverb effect is possible.*
vol	0-127	Tone volume value.
pan	0-127	Tone panning value
center	0-127	Center note (half sound units)
shift	0-127	Center note fine tuning
min	0-127	Minimum value of note reinit
max	0-127	Maximum value of note reinit
pbmin	0-127	Minimum value of pitch bend in lower direction
pbmax	0-127	Maximum value of pitch bend in lower direction
ar	0-127	Attack rate
dr	0-15	Decay rate
sr	0-127	Sustain rate
rr	0-31	Release rate
sl	0-15	Sustain level
prog	0-127	Program number attached to the tone
vag	0-254	VAG in use

* Special sound functions must be set in the reverb settings to obtain reverb effects.

Supplement

The rates for the ADSR (envelope), namely, attack, decay, sustain and release can be individually set, and by such timely changes the linear and exponential functions can be specified. It is also possible to set the sustain level.

Figure 4-7: ADSR rates



SEQ2SEP: Converts SEQ to SEP

Version 1.0

Documentation date: 3/09/96

Description

This software converts PlayStation music data format SEQ's together into SEP's. They can be collectively loaded and opened.

How to use

seq2sep [options] *SEQ-files...*

When an SEQ file is included as an argument, a PlayStation music data file SEP file (*.SEP) is created.

Options

-o Specifies the output SEP file name.

Limitations

Up to 16 SEQ's can be combined into an SEP.

VABSPLIT: Splits VAB into VH and VB

Version 2.0

Documentation date: 5/1/96

Description

This software divides the PlayStation sound source bank data (VAB) created by mkvab.exe into an attribute table section (VH) and a wave pattern data section (VB). Although the VH data must exist in the main memory when music is playing, this is not the case if the VB data is transferred to the SPU once.

How to use

vabsplit *vab-files...*

When the VAB data is included as an argument, it is divided into a PS sound source bank attribute table section and a wave pattern data section. If multiple files are included, they can be all converted together. The file extension can be shortened to [.vab].

Options

None

Constraints

None

When music is playing multiple files are included, they can all be converted together. This file extension can be shortened to VAB.

Index

3

3D Studio, 2-5, 3-57

- 3DS object, saving in PlayStation format, 3-57
- animation data, saving in PlayStation format, 3-57
- plugins
 - TOD_I.KXP, 3-57

3DFACE, 2-3, 2-4, 2-5, 2-6, 2-12, 2-15

A

Adobe Photoshop, 3-3, 3-5, 3-22, 3-23, 3-24, 3-25, 3-27, 3-28, 3-29, 3-30, 3-31, 3-32, 3-33, 3-34, 3-35, 3-36, 3-37, 3-39, 3-40, 3-41

- plugins
 - TIMEXP, 3-22, 3-27, 3-30
 - TIMexpE, 3-25
 - TIMFMT, 3-37
 - TIMfmtE, 3-27, 3-32
 - TIMformatE, 3-37

ADPCM compression, 1-6, 1-49, 1-56, 1-59

AIFF files, 4-6, 4-7

- VAG, converting to, 4-6

AIFF2VAG program, 4-6, 4-7

Alchemy™, 4-6

Alias Upfront (1.1), 2-5

Alias Wavefront

- plugins
 - MGAME.ADDIN, 3-42

animation

- hierarchical, 3-42

Artist Board, 3-25, 3-28, 3-29, 3-30

audio formats

- WAV, 1-4, 1-5, 1-21, 1-25, 1-40, 1-49, 1-58, 1-59, 1-64, 1-66
- XA, 1-3, 1-4, 1-6, 1-7, 1-8, 1-10, 1-11, 1-14, 1-15, 1-21, 1-22, 1-24, 1-25, 1-26, 1-27, 1-29, 1-31, 1-33, 1-34, 1-37, 1-38, 1-40, 1-41, 1-42, 1-43, 1-44, 1-45, 1-47, 1-49, 1-54, 1-56, 1-59, 1-60, 1-61, 1-63, 1-64, 1-65, 1-67, 1-68

AVI format, 1-4, 1-18, 1-20, 1-24, 1-25, 1-26, 1-27, 1-28, 1-32, 1-36, 1-47, 1-48, 1-49, 1-55, 1-58, 1-59, 1-60, 1-61

AVI parameters

- setting, 1-55

B

batch conversions, 1-24

binary data, 2-16

BMP files, 3-3, 3-7, 3-10, 3-12, 3-13, 3-14, 3-15, 3-17, 3-21

- multiple conversion of, 3-12
- TIM, converting from, 3-10
- TIM, converting to, 3-3

BMP2TIM program, 3-3

C

CD-ROM

- burning streaming data
 - parameters, 1-18, 1-47
- XA audio data, recording, 1-67

CLUTs, 2-32, 2-44, 3-3, 3-4, 3-5, 3-6, 3-7, 3-8, 3-11, 3-15, 3-16, 3-18, 3-19, 3-22, 3-23, 3-25, 3-27, 3-29, 3-30, 3-32, 3-33, 3-34, 3-35, 3-36, 3-37, 3-40, 3-41, 3-49, 3-56, 3-65

- autosorting, 3-6

COD files, 2-19, 2-31, 2-45

compressing, 1-13, 1-16, 1-48

conversion

- mc32.exe, 1-3
- QuickTime data, 1-47

D

decoding, 1-13, 1-14, 1-53, 1-54

display area, loading to VRAM address, 3-13, 3-22, 3-23, 3-25, 3-29

DTL-H201A, 3-22, 3-23, 3-24

- Adobe Photoshop images, viewing, 3-23

DXF files, 2-3, 2-4, 2-5, 2-6, 2-7, 2-8, 2-9, 2-10, 2-11, 2-12, 2-13, 2-14, 2-15, 2-21, 2-30

DXF format, 2-3, 2-4, 2-5, 2-6, 2-7, 2-8, 2-9, 2-10, 2-11, 2-12, 2-13, 2-14, 2-15, 2-21, 2-30

DXF2RSD program, 2-3, 2-4, 2-5, 2-6, 2-7, 2-8, 2-9, 2-11, 2-12, 2-21, 2-30, 2-40

- 3D modelers, commercial
 - convertible, 2-5
 - unconvertible, 2-6

DXF2RSDW program (Windows), 2-10, 2-11

E

encoding, 1-54

expanding, 1-16

F

form Z (2.1 and later), 2-5

functions

- SetSemiTrans(), 3-7

G

Gouraud shading, 2-5, 2-6, 2-12, 3-46

GRP files, 2-3, 2-10, 2-19, 2-31, 2-33, 2-35, 2-36, 2-46, 3-45

H

HMD data, 2-16, 2-17, 2-18, 2-19, 2-25, 2-26, 2-31, 2-32, 2-45, 2-50

- labp format, converting to, 2-50
- MIMe, making, 2-25
- RSD, converting from, 2-31

HRC files, 2-27, 2-28

I

interleaving, audio and video data, 1-3, 1-11, 1-22, 1-24, 1-49, 1-52, 1-55, 1-56, 1-59, 1-65, 1-68

J

JPEG files, 1-14, 1-48, 1-54, 3-4, 3-7, 3-12, 3-21

K

key frames, 2-24

L

LABP program, 2-16

example input file, 2-17

labp-formatted files, 2-16

HMD, converting from, 2-50

leap sectors, 1-9, 1-13, 1-19, 1-26, 1-27, 1-28, 1-29, 1-30, 1-31, 1-33, 1-34, 1-35, 1-37, 1-38, 1-39, 1-41, 1-52, 1-56, 1-60, 1-61, 1-62, 1-63, 1-65, 1-66

LightWave 3D, 3-64

plugins, 3-64

RSD geometry file loader, 3-64

RSD geometry file saver, 3-64

TIM image file loader and saver, 3-64

M

MacroModel (1.5), 2-6

MAT files, 2-3, 2-7, 2-15, 2-19, 2-31, 2-32, 2-33, 2-34, 2-35, 2-36, 2-40, 2-46, 3-28, 3-45, 3-50

mc32.exe program, 1-3

batch conversions, 1-24

compressing and expanding files, 1-7

frequency settings, 1-17

images, verifying, 1-16

MDEC compression quantization matrices, changing, 1-14

MDEC compression, setting parameters for, 1-12

MDEC videos, creation of, 1-18

multiple channels, creating, 1-10

scripts, 1-15

video/sound synthesis, performing, 1-8

XA compression parameters, setting, 1-14

MDEC, 1-3, 1-4, 1-5, 1-7, 1-8, 1-9, 1-10, 1-11, 1-12, 1-13, 1-14, 1-16, 1-18, 1-19, 1-20, 1-24, 1-25, 1-26, 1-27, 1-28, 1-29, 1-30, 1-31, 1-32, 1-33, 1-34, 1-35, 1-36, 1-37, 1-38, 1-39, 1-47, 1-48, 1-49, 1-51, 1-52, 1-53, 1-54, 1-56, 1-58, 1-59, 1-60, 1-61, 1-62, 1-63, 1-64, 1-68

parameters, 1-51

MESHCONV program, 2-19

polygon linkage, retaining, 2-19

MGAME.ADDIN program, 3-42

backfacing, 3-50

light source, 3-50

output filename, selecting, 3-50

scaling, 3-50

screen size, setting, 3-48

STDERR, outputting messages to, 3-50

textures, 3-46, 3-48, 3-49, 3-50

triangulating polygons, 3-50

Vmapper... window, 3-45

MIDI files, 4-3, 4-4

MIMe, 2-6, 2-21, 2-23, 2-24, 2-25, 2-26, 2-50

animation, 2-6, 2-23

speed, increasing, 2-23

difference file, 2-21, 2-23

MIMEFILT program, 2-21, 2-23

MIMESORT program, 2-23

MKMIME program, 2-25

MKTOD program, 2-27, 2-28, 2-40

MKVAB program, 4-9

MOT files, 2-19, 2-31, 2-32, 2-45

MovConv program, 1-41, 1-49, 1-50, 1-51, 1-54, 1-55, 1-56, 1-57, 1-58, 1-59, 1-65, 1-66, 1-68

audio conversion differences, 1-52

AVI parameters, 1-55

leap sectors, 1-9, 1-13, 1-19, 1-26, 1-27, 1-28, 1-29, 1-30, 1-31, 1-33, 1-34, 1-35, 1-37, 1-38, 1-39, 1-41, 1-52, 1-56, 1-60, 1-61, 1-62, 1-63, 1-65, 1-66

quantization matrix, 1-12, 1-14, 1-19, 1-46, 1-53, 1-54

scripting, 1-57, 1-59

XA audio parameters, 1-54

movies

tools

mc32.exe, 1-3

mc32.exe program, 1-3

MovConv program, 1-41, 1-49, 1-50, 1-51, 1-54, 1-55, 1-56, 1-57, 1-58, 1-59, 1-65, 1-66, 1-68

MovPack program, 1-67

MovPack program, 1-67

MSH files, 2-19, 2-31

multi-channel streaming, 1-67, 1-68

N

NDF files, 2-21, 3-42, 3-46

O

OGP files, 2-19, 2-31, 2-45

P

packets, 2-28, 2-47, 2-49

painting speed, 2-49

PCX files, 3-7

Performer™, 4-3

PICT files, 3-4, 3-12, 3-14, 3-15, 3-17, 3-21

multiple conversion of, 3-12

TIM, converting to, 3-4

PICT2TIM program, 3-4

PlayStation, 1-3, 1-4, 1-13, 1-16, 1-22, 1-47, 1-49, 1-52, 1-58, 2-3, 2-4, 2-5, 2-6, 2-10, 2-12, 2-17, 2-20, 2-26, 2-32, 2-36, 2-39, 2-47, 2-50, 3-3, 3-4, 3-5, 3-7, 3-9, 3-10, 3-11, 3-12, 3-14, 3-15, 3-17, 3-19, 3-21, 3-22, 3-25, 3-28, 3-32, 3-34, 3-37, 3-42, 3-44, 3-45, 3-46, 3-47, 3-48, 3-50, 3-51, 3-52, 3-53, 3-54, 3-55, 3-57, 3-60, 3-61, 3-62, 3-63, 3-64, 3-65, 4-3, 4-6, 4-11, 4-12

game console translator, 3-42

MetaCycle preview, 3-42, 3-46

PLY files, 2-3, 2-19, 2-31, 2-32, 2-33, 2-35, 2-36, 2-40, 2-44, 2-46, 3-45

PMD files, 2-47, 2-48

TMD, converting from, 2-47

POLYFACE MESH, 2-5

polygon linkage, retaining, 2-19

Polyline, 2-4, 2-5, 2-12, 2-15

PRJ files, 2-27, 2-28, 2-39, 2-42, 3-62, 3-63

PVT files, 2-19, 2-31, 2-45

Q**QuickTime data, 1-47**

conversion, 1-47

R**RGB files, 1-4, 1-5, 1-18, 1-25, 1-33, 1-34, 1-35, 1-36, 1-48, 1-49, 1-58, 1-59, 1-63, 1-64, 2-7, 2-15, 3-5, 3-7, 3-12, 3-14, 3-15, 3-17, 3-18, 3-23, 3-27, 3-29, 3-30, 3-33, 3-34, 3-35, 3-36, 3-40, 3-64, 3-65**

multiple conversion of, 3-12

TIM, converting to, 3-5

RGB format, 1-4, 1-5, 1-18, 1-25, 1-33, 1-34, 1-35, 1-36, 1-48, 1-49, 1-58, 1-59, 1-63, 1-64, 2-7, 2-15, 3-5, 3-7, 3-12, 3-14, 3-15, 3-17, 3-18, 3-23, 3-27, 3-29, 3-30, 3-33, 3-34, 3-35, 3-36, 3-40, 3-64, 3-65**RGB2TIM program, 3-5****RSD file set, 2-33****RSD files, 2-3, 2-5, 2-6, 2-7, 2-9, 2-10, 2-14, 2-15, 2-19, 2-21, 2-27, 2-28, 2-30, 2-31, 2-32, 2-33, 2-35, 2-36, 2-37, 2-38, 2-39, 2-40, 2-41, 2-43, 2-44, 2-45, 2-46, 3-13, 3-17, 3-45, 3-52, 3-57, 3-58, 3-61, 3-62, 3-63, 3-64****RSD format (PlayStation 3D model data format), 2-3, 2-5, 2-6, 2-7, 2-9, 2-10, 2-14, 2-15, 2-19, 2-21, 2-27, 2-28, 2-30, 2-31, 2-32, 2-33, 2-35, 2-36, 2-37, 2-38, 2-39, 2-40, 2-41, 2-43, 2-44, 2-45, 2-46, 3-13, 3-17, 3-45, 3-52, 3-57, 3-58, 3-61, 3-62, 3-63, 3-64**

deforming or moving data, 2-35

DXF, converting from, 2-3, 2-10

DXF, converting to, 2-30

HMD, converting to, 2-31

multiple files, linking, 2-33

old RSD files, creating HMD-compatible files from, 2-45

TMD, converting to, 2-39

RSD2DXF program, 2-30**RSD2HMD program, 2-31****RSDCAT program, 2-33****RSDFORM program, 2-27, 2-35, 2-36, 2-37, 2-40****RSDLINK program, 2-21, 2-27, 2-36, 2-39, 2-40, 2-41, 2-42, 2-43, 2-44, 3-61****RSDUP program, 2-45****S****script files, 1-15, 1-17, 1-24, 1-59, 3-53****scripting functions**

Avi2bsAv(), 1-25, 1-27

Avi2bsC(), 1-25, 1-28

Avi2bsV(), 1-25, 1-28

Avi2strMdecAv(), 1-25, 1-26, 1-59, 1-60

Avi2strMdecC(), 1-25, 1-27, 1-59, 1-61

Avi2strMdecV(), 1-25, 1-26, 1-59, 1-60

MakeAv(), 1-25, 1-40, 1-41, 1-59, 1-65, 1-66

MakeV(), 1-25, 1-41, 1-59, 1-66

Pack16ch(), 1-25, 1-45

Pack1ch(), 1-25, 1-42

Pack2ch(), 1-25, 1-42, 1-43

Pack32ch(), 1-25, 1-45, 1-46

Pack4xh(), 1-25

Pack8ch(), 1-25, 1-44

Pcm2xa(), 1-25, 1-40, 1-59, 1-65

Rgb2Avi(), 1-25, 1-36, 1-59

Rgb2bsAv(), 1-25, 1-34, 1-35

Rgb2bsC(), 1-25, 1-36, 1-59

Rgb2bsV(), 1-25, 1-35, 1-59

Rgb2strMdecAv(), 1-25, 1-33, 1-59, 1-63

Rgb2strMdecC(), 1-25, 1-34, 1-59, 1-64

Rgb2strMdecV(), 1-25, 1-33, 1-34, 1-59, 1-63, 1-64

SetDefaultQ(), 1-25, 1-47

SetQ(), 1-25, 1-46

Tim2bsAv(), 1-25, 1-38

Tim2bsC(), 1-25, 1-39, 1-59

Tim2bsV(), 1-25, 1-39, 1-59

Tim2strMdecAv(), 1-25, 1-37

Tim2strMdecC(), 1-25, 1-38

Tim2strMdecV(), 1-25, 1-37

Wav2xa(), 1-25, 1-40, 1-59, 1-64

Yuv2Avi(), 1-25, 1-32

Yuv2bsAv(), 1-25, 1-31

Yuv2bsC(), 1-25, 1-32, 1-59

Yuv2bsV(), 1-25, 1-31, 1-32, 1-59

Yuv2strMdecAv(), 1-25, 1-29, 1-59, 1-61

Yuv2strMdecC(), 1-25, 1-30, 1-59, 1-62

Yuv2strMdecV(), 1-25, 1-30, 1-59, 1-62

Sculptura (1.1), 2-5**SEP files, 4-11**

SEQ, converting from, 4-11

SEQ files, 4-3, 4-11

SEP, converting to, 4-11

SEQ format (sequence data format), 4-3, 4-11**SEQ2SEP program, 4-11****sequence data, creating, 4-3****ShadeIII (1.0), 2-6****SMF format, 4-3, 4-4****SMF2SEQ program, 4-3, 4-4****Sound Artist Tool (DTL-S710), vii****SoundDesigner II™, 4-6****STP (SemiTransparent information), 3-7, 3-8, 3-22, 3-25, 3-27, 3-29, 3-32, 3-33, 3-35, 3-37, 3-49, 3-55**

bits, 3-7, 3-8, 3-27, 3-29, 3-33, 3-35, 3-49, 3-55

controlling, 3-42, 3-49

Strata Studio Pro (1.0), 2-5**StrataVision, 2-5****streaming**

multi-channel, 1-67, 1-68

streaming data, 1-3, 1-11, 1-13, 1-21, 1-22, 1-23, 1-24, 1-47, 1-48, 1-52, 1-67, 1-68

managing

MovPack program, 1-67

streaming format (STR), 1-4, 1-5, 1-8, 1-9, 1-10, 1-11, 1-16, 1-19, 1-24, 1-25, 1-26, 1-27, 1-29, 1-30, 1-33, 1-34,

1-37, 1-38, 1-42, 1-43, 1-44, 1-45, 1-47, 1-49, 1-51, 1-56, 1-58, 1-59, 1-60, 1-61, 1-62, 1-63, 1-64, 1-68

Streaming format (STR)

MDEC, 1-3, 1-4, 1-5, 1-7, 1-8, 1-9, 1-10, 1-11, 1-12, 1-13, 1-14, 1-16, 1-18, 1-19, 1-20, 1-24, 1-25, 1-26, 1-27, 1-28, 1-29, 1-30, 1-31, 1-32, 1-33, 1-34, 1-35, 1-36, 1-37, 1-38, 1-39, 1-47, 1-48, 1-49, 1-51, 1-52, 1-53, 1-54, 1-56, 1-58, 1-59, 1-60, 1-61, 1-62, 1-63, 1-64, 1-68

T

texture CLUT, loading to VRAM address, 3-22, 3-23, 3-25, 3-27, 3-29, 3-32, 3-37, 3-39

texture mapping, 3-16, 3-22, 3-25, 3-31, 3-64

texture page, loading to VRAM address, 3-20, 3-22, 3-23, 3-25, 3-29, 3-32, 3-37, 3-39

TIFF files, 3-7

TIM format, 1-4, 1-5, 1-25, 1-37, 1-38, 1-39, 1-49, 1-58, 1-59, 2-28, 2-31, 2-32, 2-40, 2-45, 2-46, 3-3, 3-4, 3-5, 3-6, 3-7, 3-8, 3-9, 3-10, 3-11, 3-12, 3-13, 3-14, 3-15, 3-16, 3-17, 3-18, 3-19, 3-21, 3-22, 3-23, 3-27, 3-28, 3-31, 3-32, 3-33, 3-34, 3-35, 3-36, 3-37, 3-38, 3-39, 3-40, 3-41, 3-45, 3-46, 3-52, 3-53, 3-55, 3-56, 3-64, 3-65

Adobe Photoshop, exporting images, 3-32, 3-37

Adobe Photoshop, importing images, 3-32, 3-37

BMP, converting from, 3-3, 3-7

BMP, converting to, 3-10

JPEG, converting from, 3-7

load time, speeding up, 3-9

multiple conversion of, 3-12

pallet (CLUT) addressess, changing, 3-11

PCX, converting from, 3-7

PICT, converting from, 3-4

RGB, converting from, 3-5

STP bit, controlling, 3-42, 3-49

TIFF, converting from, 3-7

TIMTOOL, 3-6

VRAM, changing, 3-11

TIM2BMP program, 3-10

TIMEXP program, 3-22, 3-27, 3-30

TIMexpE program, 3-25

TIMFMT program, 3-37

TIMfmtE program, 3-27, 3-32

TIMformatE program, 3-37

TIMPOS program, 3-11, 3-17, 3-21

TIMTOOL program, 3-6

TIMUTIL program, 3-12

pallet for display, setting, 3-16

parameter settings, 2-11, 3-14

screen resolution, setting, 3-16

VRAM position, setting, 3-16

TMD files, 2-21, 2-23, 2-24, 2-25, 2-27, 2-28, 2-36, 2-39, 2-40, 2-41, 2-42, 2-43, 2-44, 2-47, 2-48, 2-49, 3-42, 3-45, 3-46, 3-50, 3-52, 3-58, 3-60, 3-61

PMD, converting to, 2-47

sorting and displaying, 2-49

vertex information, sorting, 2-23

TMD2PMD program, 2-47

TMDINFO program, 2-49

TMDSORT program, 2-49

TOD files, 2-27, 2-28, 2-36, 2-40, 3-42, 3-45, 3-46, 3-50, 3-52, 3-55, 3-57, 3-58, 3-59, 3-60, 3-61, 3-62, 3-63

TOD_I.KXP program, 3-57

trueSpace (1.0), 2-5

V

VAB data, 4-9, 4-12

VH and VB, splitting into, 4-12

VAB files, 4-9, 4-12

VABSPLIT program, 4-12

VAG files, 4-6, 4-7, 4-9, 4-10

AIFF, converting from, 4-6

VAG format, 4-6, 4-7, 4-9, 4-10

VB data, 4-12

VDF files, 2-21, 2-22, 3-42, 3-46

vertex information, sorting, 2-23

VH data, 4-12

video capture board, 1-18, 1-48

video data conversion

MovConv program, 1-41, 1-49, 1-50, 1-51, 1-54, 1-55, 1-56, 1-57, 1-58, 1-59, 1-65, 1-66, 1-68

Vision™, 4-3

W

WAV audio format, 1-4, 1-5, 1-21, 1-25, 1-40, 1-49, 1-58, 1-59, 1-64, 1-66

waveform data, creating, 4-6

Windows 3.1, 1-3

Windows 95, 1-3, 3-25, 3-32, 3-52

XA audio format, 1-3, 1-4, 1-6, 1-7, 1-8, 1-10, 1-11, 1-14, 1-15, 1-21, 1-22, 1-24, 1-25, 1-26, 1-27, 1-29, 1-31, 1-33, 1-34, 1-37, 1-38, 1-40, 1-41, 1-42, 1-43, 1-44, 1-45, 1-47, 1-49, 1-54, 1-56, 1-59, 1-60, 1-61, 1-63, 1-64, 1-65, 1-67, 1-68

parameters, 1-54

X

XHMD program, 2-50

Y

YUV format, 1-4, 1-5, 1-18, 1-25, 1-29, 1-30, 1-31, 1-32, 1-48, 1-49, 1-58, 1-59, 1-61, 1-62