

Movie Converter mc32

This document explains the movie converter (mc32), a Windows 95 / Windows NT-compatible video compression and conversion tool.

Contents

1 Movie Tool.....	3
2 mc32 Overview.....	4
Features.....	4
Functions	4
Supported Formats.....	4
3 mc32 Operation.....	7
Main Window	7
Conversion Window	8
Video+Sound Window	9
Multi Channel Window	11
MDEC Parameters Window	13
MDEC Parameter - Q Matrix Window	16
XA Parameters Window	17
Script Window	18
Video Window.....	19
Sound Window	20
General Setting Dialog Box.....	20
Data Conversion without Using a GUI.....	21
4 Creating Video Data.....	22
Video Creation Procedure	22
Creating Video Data Without Sound.....	23
Creating Video Data with Sound (Simultaneous Conversion of Sound).....	24
Creating Video Data with Sound (Separate Conversion of Sound).....	25
5 Creating Sound Data	26
6 Creating Multichannel Data	28
Multichannel Streaming	28
Creating Multichannel Data.....	29
7 Scripting	32
Script Function Reference	33
8 Miscellaneous	55
QuickTime Data Conversion	55
Burning Streaming Data onto a CD-ROM	55
Capturing and Compressing High-Image-Quality Video	56

1 Movie Tool

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 known as 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.exe and movpack.exe are available for Windows 3.1 and Windows 95, however, the new movie converter mc32.exe, described in this document, integrates these functions and provides much faster image compression by comparison. mc32.exe runs under Windows 95 and Windows NT.

This document describes the functionality and use of mc32.exe by way of actual examples. For information regarding the previous movie tools, movconv.exe and movpack.exe, please refer to the documents movconv.doc and movpack.doc.

Table 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 (Windows3.1/95)
Movpack.exe	Multichannel interleave (Windows3.1/95)

2 mc32 Overview

Features

mc32 provides video compression and sound compression, format conversion, multiple channel creation, video/sound synthesis, 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 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.

3 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

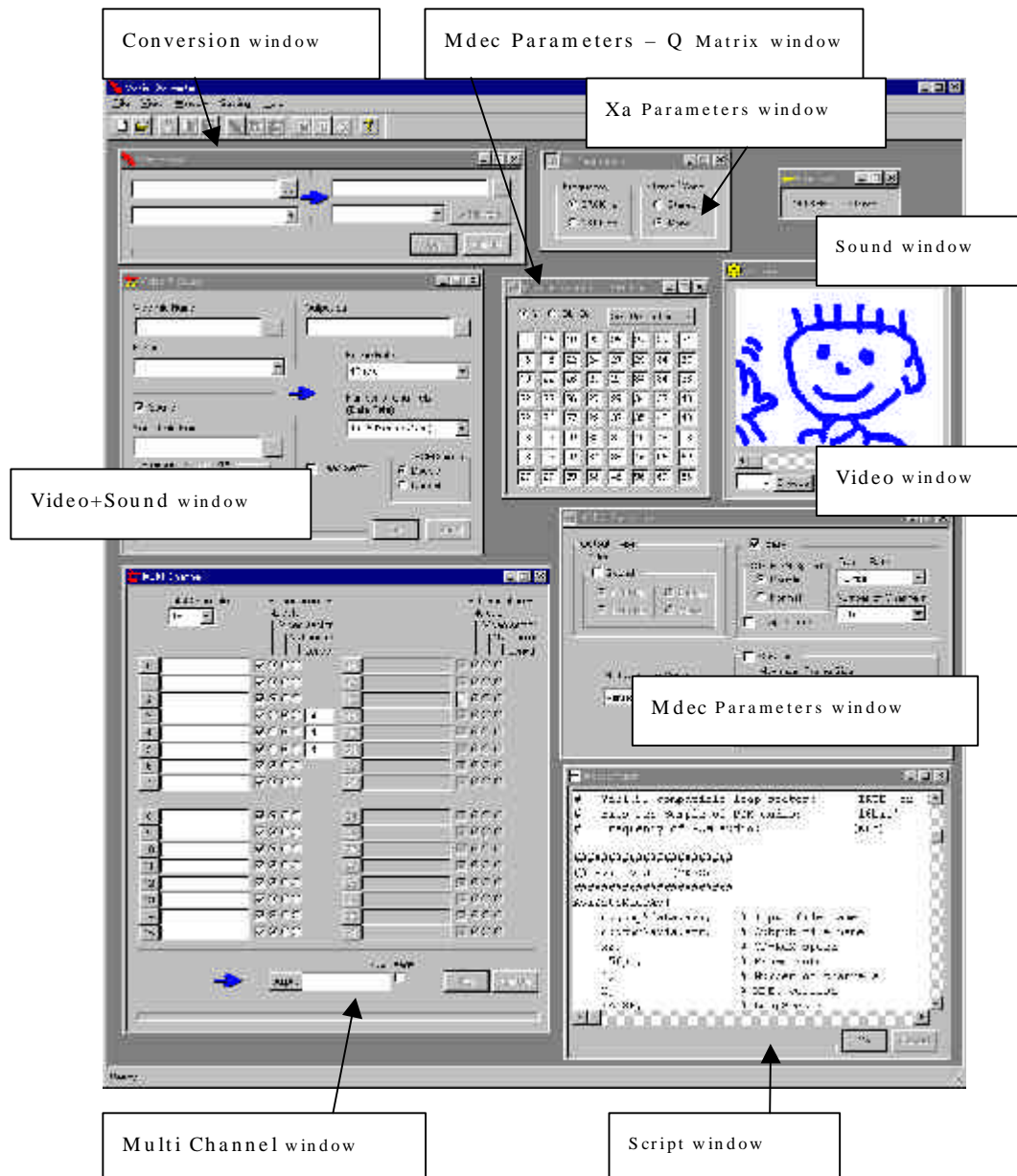


Figure 1 Main window

Figure 1 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

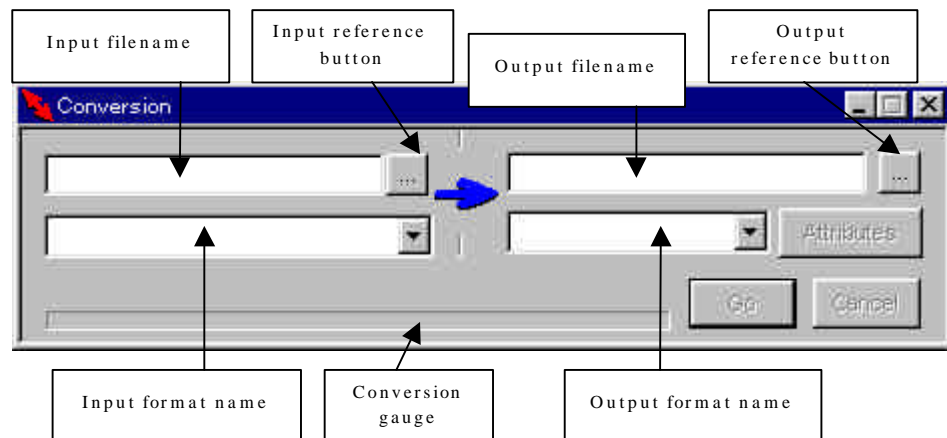


Figure 2 Conversion window

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

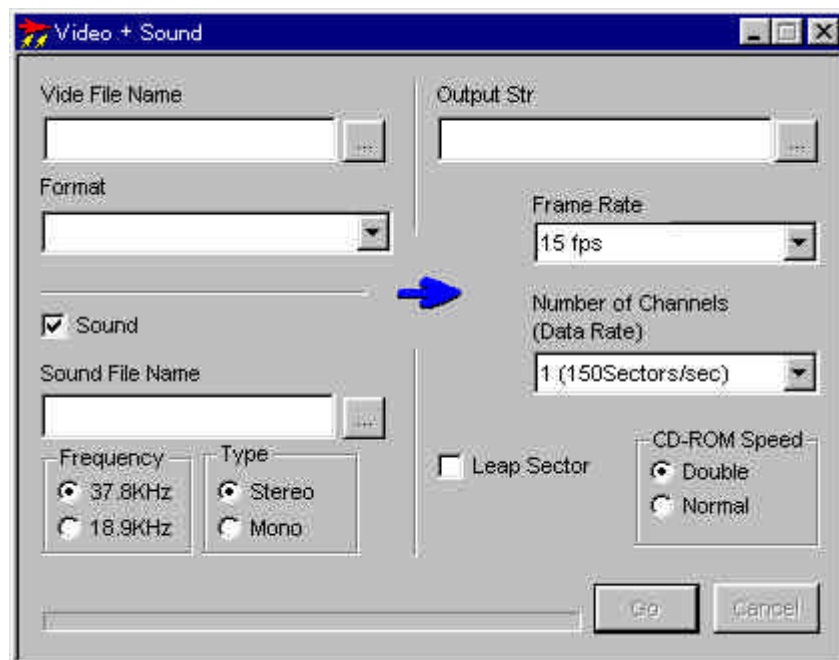


Figure 3 Video+Sound window

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

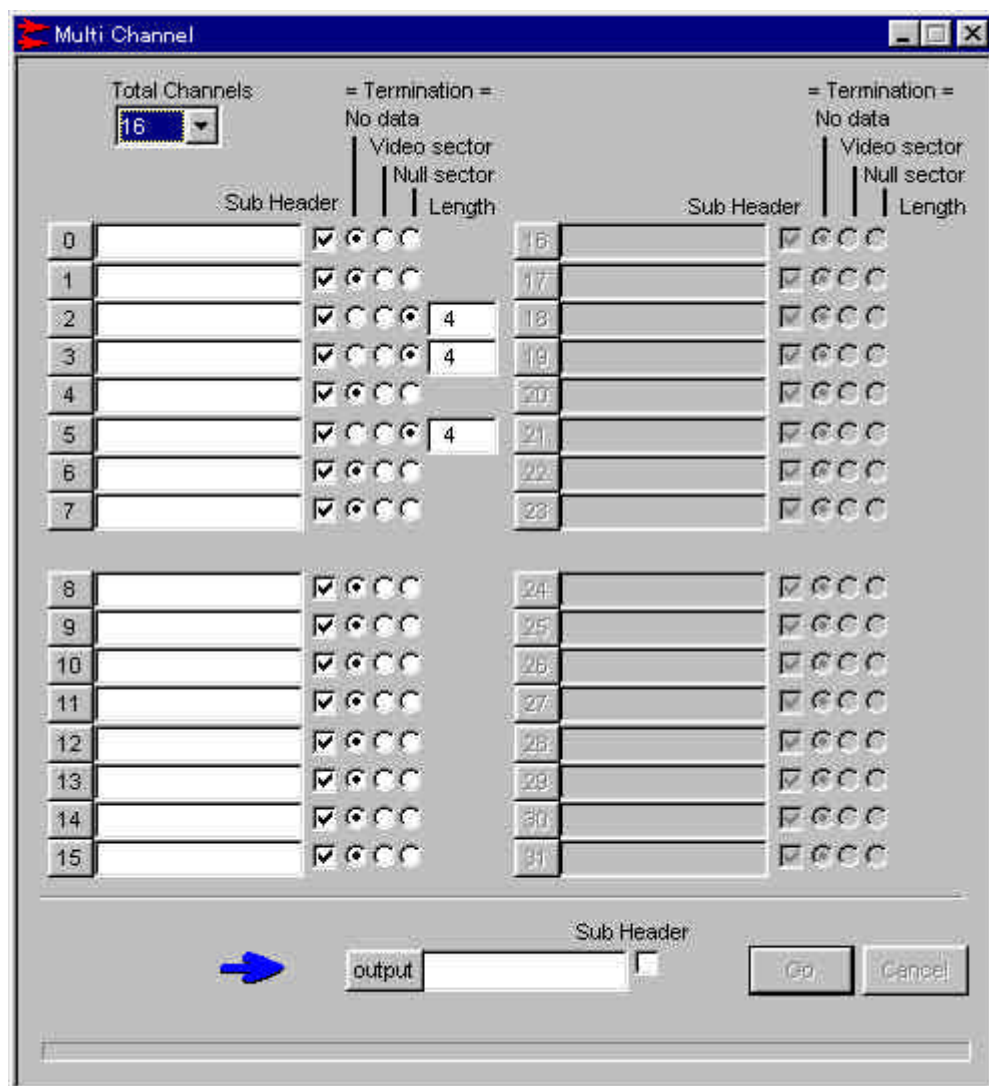


Figure 4 Multi Channel window

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 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

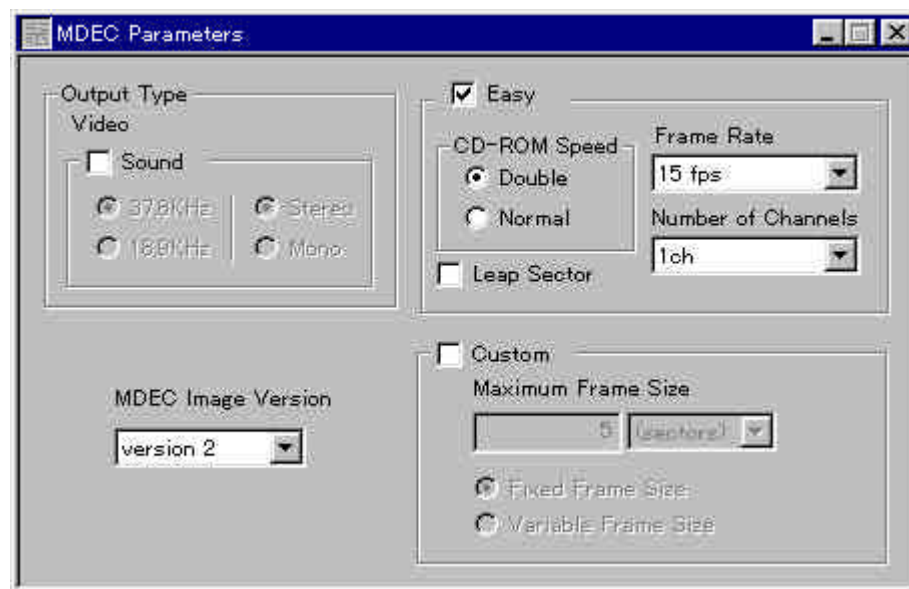


Figure 5 MDEC Parameters window

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 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 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 sector = 2016 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

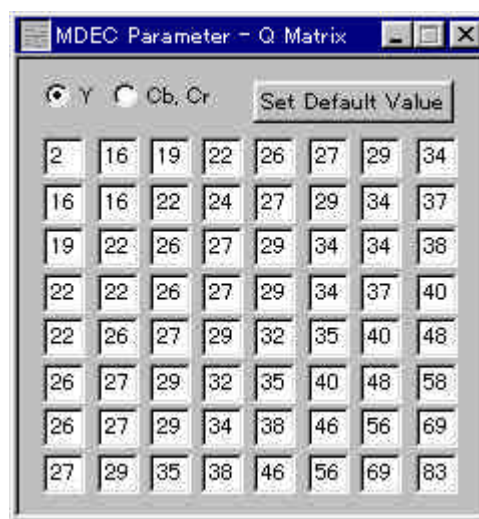


Figure 6 MDEC Parameter - Q Matrix window

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

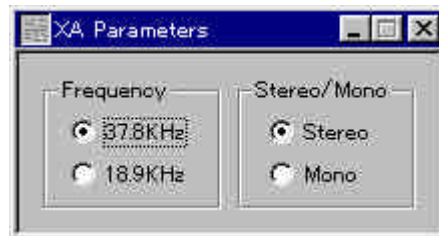


Figure 7 XA Parameters window

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

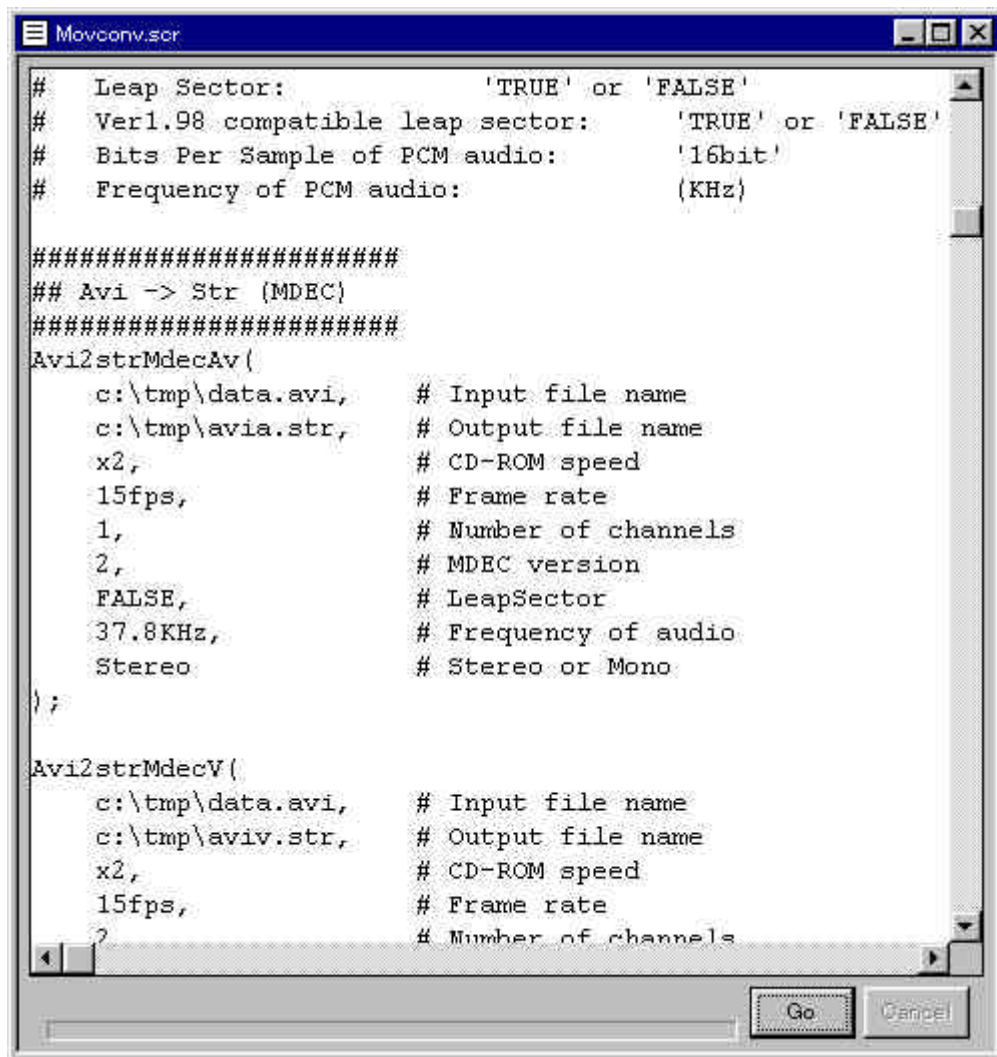


Figure 8 Script window

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

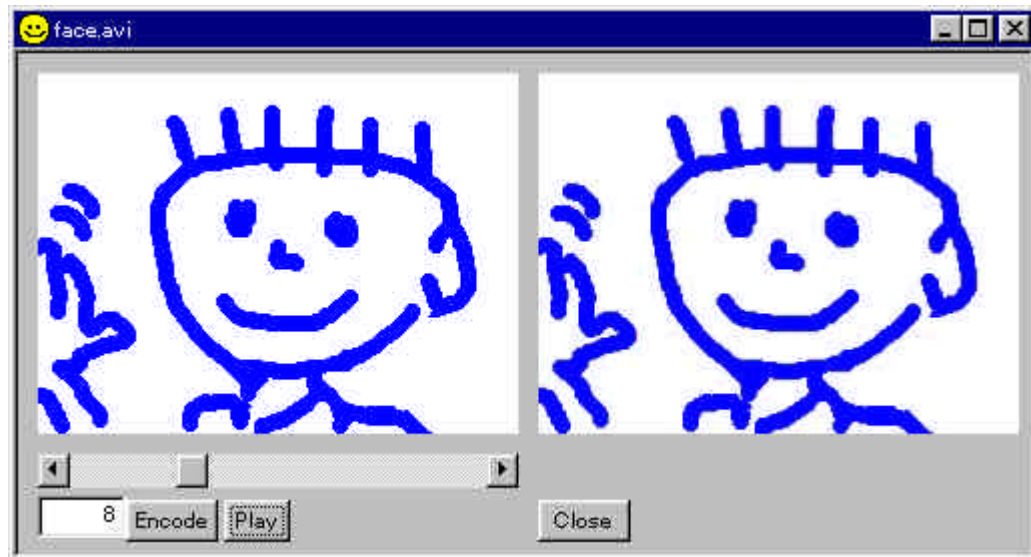


Figure 9 Video window

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

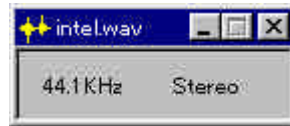


Figure 10 Sound window

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

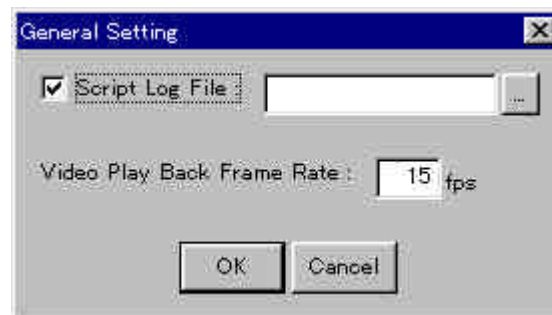


Figure 11 General Setting dialog box

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.

4 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.

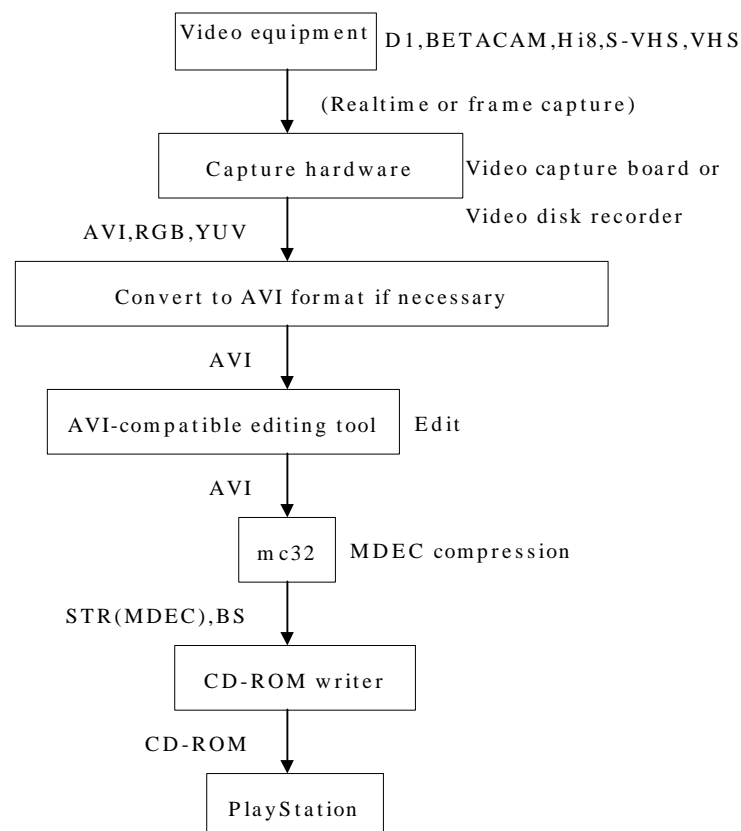


Figure 12 Video data creation procedure

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.

5 Creating Sound Data

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

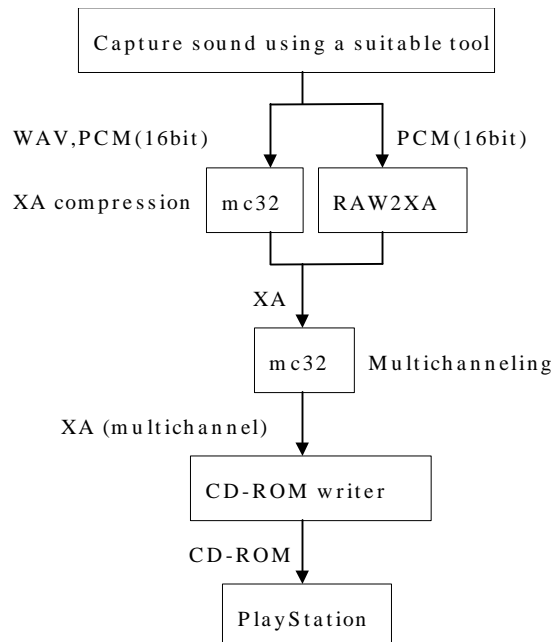


Figure 13 Sound data creation procedure

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.

6 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.

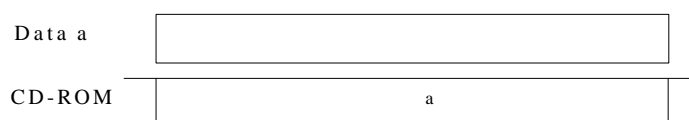


Figure 14 Ordinary streaming

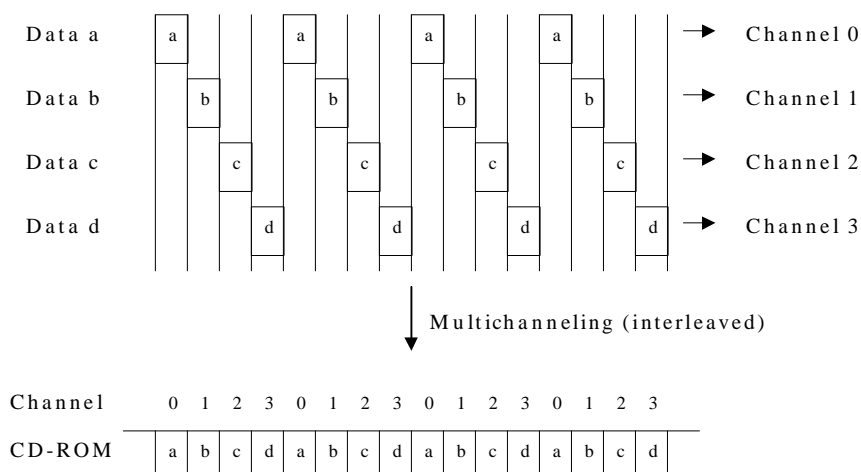


Figure 15 Multichannel streaming

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.

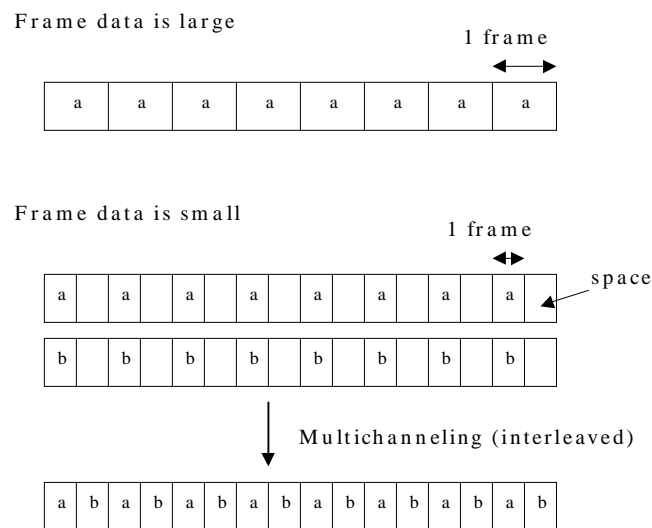


Figure 16 Efficient use of CD-ROM storage with multiple channels

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.

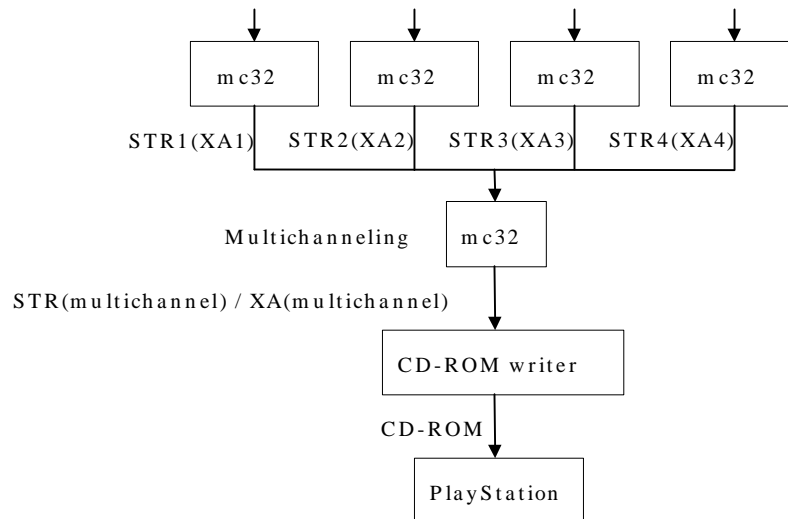


Figure 17 Procedure for creating multichannel data

(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.

7 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 6 Script Functions

Function	Description
<code>Avi2strMdecAv()</code>	Creates STR(MDEC) from AVI (Easy Settings).
<code>Avi2strMdecV()</code>	Creates STR(MDEC) from AVI (Easy Settings).
<code>Avi2strMdecC()</code>	Creates STR(MDEC) from AVI (Custom Settings).
<code>Avi2bsAv()</code>	Creates BS from AVI (Easy Settings).
<code>Avi2bsV()</code>	Creates BS from AVI (Easy Settings).
<code>Avi2bsC()</code>	Creates BS from AVI (Custom Settings).
<code>Yuv2strMdecAv()</code>	Creates STR(MDEC) from YUV (Easy Settings).
<code>Yuv2strMdecV()</code>	Creates STR(MDEC) from YUV (Easy Settings).
<code>Yuv2strMdecC()</code>	Creates STR(MDEC) from YUV (Custom Settings).
<code>Yuv2bsAv</code>	Creates BS from YUV (Easy Settings).
<code>Yuv2bsV</code>	Creates BS from YUV (Easy Settings).
<code>Yuv2bsC</code>	Creates BS from YUV (Custom Settings).
<code>Yuv2Avi</code>	Creates AVI from YUV.
<code>Rgb2strMdecAv()</code>	Creates STR(MDEC) from RGB (Easy Settings).
<code>Rgb2strMdecV()</code>	Creates STR(MDEC) from RGB (Easy Settings).
<code>Rgb2strMdecC()</code>	Creates STR(MDEC) from RGB (Custom Settings).
<code>Rgb2bsAv</code>	Creates BS from RGB (Easy Settings).
<code>Rgb2bsV</code>	Creates BS from RGB (Easy Settings).
<code>Rgb2bsC</code>	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).
Pack4xh	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.

Script Function Reference

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

[Function] Creates STR(MDEC) 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)
SoundFrequency:	Selects the XA sound frequency(37.8KHz/18.9KHz)
StereoMono:	Specifies stereo or monaural.(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 Sound
Stereo                # Stereo or Mono
);

```

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

[Function] Creates STR(MDEC) 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]

```

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)

[Function] Creates STR(MDEC) 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]

```

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
);

```

Avi2bsAv(InputFilename, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[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)
SoundFrequency:	Selects the XA sound frequency(37.8KHz/18.9KHz)
StereoMono:	Specifies stereo or monaural.(Stereo/Mono)

[Example]

```
Avi2bsAv(
    c:\tmp\data.avi,          # Input file name
    c:\tmp\avia0000.bs,      # 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
);
```

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
);

```

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

[Function] Creates STR(MDEC) from YUV (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]

```

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 Sound
Mono              # Stereo or Mono
);

```

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

[Function] Creates STR(MDEC) from YUV (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]

```

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)

[Function] Creates STR(MDEC) 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]

```
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
);
```

Yuv2bsAv (InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[Function] Creates BS from YUV (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]

```
Yuv2bsAv(
    c:\tmp\data0000.yuv, # Input file name
    320,                  # Width
    240,                  # Height
    c:\tmp\yuva0000.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
);
```

Yuv2bsV(InputFilename, Width, Height, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector)

[Function] Creates BS from YUV (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]

```
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\yuvv0000.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
);
```

Tim2strMdecC(InputFilename, OutputFilename, MaxFrameSize,FixedFrameSize, MDECversion)

[Function] Creates STR(MDEC) 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]

```
Tim2strMdecC(
    c:\tmp\data0000.tim, # 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
);
```

Tim2bsAv(InputFilename, OutputFilename, CD-ROMspeed, FrameRate, NumberOfChannels, MDECversion, LeapSector, SoundFrequency, StereoMono)

[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)

SoundFrequency: Selects the XA sound frequency(37.8KHz/18.9KHz)

StereoMono: Specifies stereo or monaural.(Stereo/Mono)

[Example]

```
Tim2bsAv(
    c:\tmp\data0000.tim, # Input file name
    c:\tmp\tima0000.bs, # 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
);
```

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] ynthesizes 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
);
```

**Pack1ch(OutputFilename, OutputSubheader,
ChFileName, ChSubheader, ChTermType, ChTermLength
)**

[Function] Creates multiple channels (1 ch) of STR data or XA data.
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)
ChFileName:	Specifies the filename of the channel data. Specify 'none' for an empty channel.
ChSubheader:	Sets the subheader attribute of the channel data.(TRUE/FALSE)
ChTermType:	Specifies the channel terminator type. (0:No data/1:Video sector/2:Null sector)
ChTermLength:	Specifies the length of the channel terminator data. This value is significant only when ChTermType is 1 or 2. (unit: sector)

[Example]

```
Pack1ch(
    # 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\data.str,    FALSE,                0,                0
);

```

**Pack2ch(OutputFilename, OutputSubheader,
ChFileName0, ChSubheader0, ChTermType0, ChTermLength0,
ChFileName1, ChSubheader1, ChTermType1, ChTermLength1
)**

[Function] Creates multiple channels of STR data or XA data.(2ch)
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)
ChFileName0-1:	Specifies the filename for each channel data. ‘None’ is specified for an empty channel.
ChSubheader0-1:	Sets the subheader attribute for each channel data.(TRUE/FALSE)
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,
    .....
    d:\tmp\data7.xa,      FALSE,             0,

);
```

Pack16ch(OutputFilename, OutputSubheader,
ChFileName0, ChSubheader0, ChTermType0, ChTermLength0,
.....
ChFileName15, ChSubheader15, ChTermType15, ChTermLength15
)

[Function] Creates multiple channels of STR data or XA data.(16ch)
 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)

ChFileName0-15: Specifies the filename for each channel data.
'None' is specified for an empty channel.

ChSubheader0-15: Sets the subheader attribute for each channel data.(TRUE/FALSE)

ChTermType0-15: Specifies the terminator type for each channel.
(0:No data/1:Video sector/2:Null sector)

ChTermLength0-15: Specifies the data length for each channel's terminator data.
This value is significant only when ChTermType is 1 or 2.
(unit: sector)

[Example]

```
Pack16ch(
    # 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\data15.xa,     FALSE,              0,              0
);

```

**Pack32ch(OutputFilename, OutputSubheader,
ChFileName0, ChSubheader0, ChTermType0, ChTermLength0,
.....
ChFileName31, ChSubheader31, ChTermType31, ChTermLength31
)**

[Function] Creates multiple channels of STR data or XA data.(32ch)
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)
ChFileName0-31:	Specifies the filename for each channel data. 'None' is specified for an empty channel.
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);

8 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:

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

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.