Multimedia File Conversion Tools

User Guide

May 2007

05-2453-003

About This Publication

Purpose

This document provides information on how to use the off-line multimedia file conversion tools (also referred to as utilities). These tools can be used to convert multimedia file data between industry-standard formats and the proprietary format.

The proprietary multimedia format is used by the Multimedia API, which provides the ability to record and play back digitized multimedia (audio and video) to support video services in application programs.

Applicability

The information in this version of the document (05-2453-003) is originally published for Dialogic® Multimedia Platform for AdvancedTCA Software Release 1.1 Service Update (also referred to as Multimedia Platform).

This document may also apply to later releases of the Multimedia Platform as well as Host Media Processing software releases. Check the *Release Guide* and the *Release Update* for your product release to determine whether this document is supported.

How to Use this Publication

The information in this publication is organized as follows:

- Section 1, “Product Description”, on page 4 – provides a brief description of the multimedia file conversion tools, where to obtain them, and the proprietary multimedia format.
- Section 2, “The mmconvert Utility”, on page 5 – provides detailed information for using the mmconvert utility.
- Section 3, “The hmp3gp Utility”, on page 7 – provides detailed information for using the hmp3gp utility.
- Section 4, “Proprietary Video File Format”, on page 9 – describes details of the proprietary video file format.
- Section 5, “Proprietary Native Audio File Format”, on page 11 – describes details of the proprietary native audio file format.

Related Information

Product documentation is available in the on-line documentation bookshelf provided with the software release or from the following support web site:
http://www.dialogic.com/support/helpweb

For information on the Multimedia API, see the following documents in the bookshelf for your product release:

- For information on Multimedia API features, as well as an overview of video technology and guidelines for developing applications using the Multimedia API, see the *Multimedia API Programming Guide*.
- For reference information on all functions, parameters, data structures, values, events, and error codes in the Multimedia API, see the *Multimedia API Library Reference*.
- For information about the Multimedia API demo, see the *Multimedia for Host Media Processing Demo Guide*.

Revision History

This revision history summarizes the changes made in each published version of this document.

Document No.	Publication Date	Description of Revisions
05-2453-003	May 2007	Product Description section: Added proprietary native audio file format in Proprietary Multimedia Format section. The mmconvert Utility section: Added mm08 option for -a audio-version parameter in mmconvert Parameters table. The hmp3gp Utility section: Added -a audio-version parameter. Updated usage example. New Proprietary Native Audio File Format section: Added to provide details on this file format.

Document No.	Publication Date	Description of Revisions
05-2453-002	July 2006	Global changes: Corrected document title references that referred only to the Linux operating system.
05-2453-001	January 2006	Initial version of document. Much of the information contained in this document was previously published in Chapter 5 of the *Multimedia API for Linux Programming Guide*, document number 05-2455-001. The file conversion information from the *Programming Guide* was moved to this document and updated here with the following revisions. Global changes: Made general improvements to descriptive information and organization. The hmp3gp Utility section: Based on improvements to the utility, changed feature description to indicate conversion in both directions, and added new -d option to the command line interface for specifying the direction of the conversion between input (source) and output (destination) files. How to Obtain the Tools section: Based on web site revisions, changed the URL for the web page used to obtain the tools.

1 Product Description

1.1 Multimedia File Conversion Tools Provided

This document provides information on the following multimedia file conversion tools:

mmconvert
: Converts multimedia file data between AVI type 2 format and the proprietary format. For details of the features and operation, see Section 2, “The mmconvert Utility”, on page 5.

hmp3gp
: Converts multimedia file data between 3rd Generation Partnership Project (3GPP) Release 4 format and the proprietary format. For details of the features and operation, see Section 3, “The hmp3gp Utility”, on page 7.

1.2 How to Obtain the Tools

The conversion tools can be downloaded from the following web site. Check this web site periodically for updates to the conversion tools and their capabilities, and for corresponding updates to this document:

http://www.dialogic.com/support/helpweb/dxall/hmpmedia/default.htm

1.3 Proprietary Multimedia Format

The proprietary multimedia format is used by the Multimedia API for video and audio play and record operations. It consists of a separate video file and one of the audio file formats, as follows:

- **Proprietary Video File Format:** Details of the format are described in Section 4, “Proprietary Video File Format”, on page 9.
- **Proprietary Audio File Format:** Linear PCM (128 kbps); 16-bit, 8 kHz, mono, LSB-MSB (“little-endian”).
- **Proprietary Native Audio File Format:** Details of the format are described in Section 5, “Proprietary Native Audio File Format”, on page 11.

2 The mmconvert Utility

This section provides detailed information for using the mmconvert utility. The following topics are covered:

2.1 Features of the mmconvert Utility

The mmconvert utility converts multimedia data between the following file formats:

- Audio Video Interleave (AVI) Type-2 format with DVSD and DV25 encoded video (in PAL 720 x 576 or NTSC 720 x 480 video format) and with PCM linear 16-bit (any rate) mono or stereo audio format
- Proprietary multimedia format using an audio and a video file

Note: The mmconvert utility can convert files from AVI file format to the proprietary multimedia file format. It does not convert from the proprietary multimedia file format to AVI file format.

Conversion includes selectable picture format (sub-QCIF, QCIF, CIF), aspect ratio adaptation, bit rate, and frames per second (30, 15, 10, 6).

2.2 mmconvert Command Line Interface

This section describes the command line interface for the mmconvert utility.

Note: The conversion utilities perform CPU-intensive tasks and should only be used when sufficient CPU capacity is available and when they will not affect other operations on the system. For example, they should not be used while performing audio/video operations or when processing audio/video calls, because this can impact the performance and operation of the system.

Command Line Parameters

```
mmconvert
    <AVI-filename>
    -a<audio-version>
    -v<video-version>
    -z<picture-format>:<adaptation-mode>:<bit-rate>:<fps>
    -y<parameter-file>
    -x<output-video-filename>
    -w<output-audio-filename>
```

Table 1. mmconvert Parameters

	Parameter	Description
	input-filename	specifies filename for AVI format file used for input
-a	audio-version	specifies version of proprietary audio file format used for output. Set to one of the following: • mm07 – linear PCM (128 kbps) proprietary audio file format • mm08 – proprietary native audio file format. For details, see Section 5, “Proprietary Native Audio File Format”, on page 11.
-v	video-version	specifies version of proprietary video file format used for output. Set to version mm07. For details on the video file format, see Section 4, “Proprietary Video File Format”, on page 9.
-z	picture-format	specifies the output picture format. Set to one of the following: • CIF – Common Intermediate Format (CIF) picture size (PAL 352 pixels by 288 pixels) • QCIF – Quarter Common Intermediate Format (QCIF) picture size (PAL 176 pixels by 144 pixels) • SQCIF – Sub-QCIF picture size (PAL 128 pixels by 96 pixels), used for mobile handsets
	adaptation-mode	specifies mode for picture adaptation. Set to one of the following: • 0 – keep original aspect ratio, cut sides, fill top and bottom (valid for CIF and QCIF only) • 2 – fit to target aspect ratio by stretching or shrinking
	bit-rate	specifies video bit rate (kbps) for output file
	fps	specifies video frames per second (30, 15, 6) for output file
-y	parameter-filename	specifies parameter file name. The file contains internal configuration parameters required for multimedia conversion algorithms, and should not be modified under normal circumstances.

Table 1. mmconvert Parameters (Continued)

	Parameter	Description
-x	output-video-filename	specifies file name of proprietary format video file used for output
-w	output-audio-filename	specifies file name of proprietary format audio file used for output

Usage Example

Command Line:

```
mmconvert  /src/demo.avi  -amm07  -vmm07  -zCIF:0:200:30
           -y  ./h263.par  -x  /dst/demo.vid  -w  /dst/demo.aud
```

This command takes the AVI file (*demo.avi*) and produces the proprietary format video file (*demo.vid*) and audio file (*demo.aud*) as output. The resulting *demo.vid* video file is produced as CIF (352 x 288 pixels), in adaptation mode 0, and is suitable for transmission over 200 kbps line at 30 frames per second.

3 The hmp3gp Utility

This section provides detailed information for using the hmp3gp utility. The following topics are covered:

3.1 Features of the hmp3gp Utility

The hmp3gp utility converts multimedia data between the following file formats:

- 3rd Generation Partnership Project (3GPP) Release 4 file format (.3gp)
- HMP software proprietary multimedia format using an audio and a video file

Note: The hmp3gp utility can convert files in either direction between 3GP format and proprietary multimedia format.

The 3GP file contains two tracks: a video track with H.263 bit-stream video data, and an audio track with Global System for Mobile communication Adaptive Multi-Rate Narrow Band (GSM-AMR-NB) audio encoded at a bitrate of 12.2 kbps.

Note: No transrating or resizing is done, so the destination frame rate and the picture size will be the same as the source.

3.2 hmp3gp Command Line Interface

This section describes the command line interface for the hmp3gp utility.

Note: The conversion utilities perform CPU-intensive tasks and should only be used when sufficient CPU capacity is available and when they will not affect other operations on the system. For example, they should not be used while performing audio/video operations or when processing audio/video calls, because this can impact the performance and operation of the system.

Command Line Parameters

```
hmp3gp
        -d<direction>
        -a<audio-version>
        <Intel-video-filename>
        <Intel-audio-filename>
        <3gp-filename>
```

Table 2. hmp3gp Parameters

	Parameter	Description
-d	direction	specifies direction or destination of conversion between the specified files. The sequence of file names in the command line stays the same, and the direction parameter indicates which files are used as source and which as destination for the conversion. Set to one of the following: • 0 – convert to 3GP format (default) • 1 – convert to Intel proprietary multimedia format
-a	audio-version	specifies version of proprietary audio file format used for output. Set to one of the following: • mm07 – linear PCM (128 kbps) proprietary audio file format • mm08 – proprietary native audio file format. For details, see Section 5, “Proprietary Native Audio File Format”, on page 11.
	Intel-video-filename	specifies file name of proprietary format video file
	Intel-audio-filename	specifies file name of proprietary format audio file
	3gp-filename	specifies file name for 3GP format file

Usage Example

Command Line:

```
hmp3gp  -d0 -amm07 /src/demo.vid  /src/demo.aud  /dst/demo.3gp
```

This command takes the proprietary format video file (*demo.vid*) and audio file (*demo.aud*) and produces the 3GP format file (*demo.3gp*) as output. This is the default conversion direction if the -d parameter is not specified.

If -d1 were specified, it would take the 3GP file and produce the proprietary format audio and video files from it.

4 Proprietary Video File Format

This section describes the proprietary video file format.

```
Version of this File: 0.7


General: Video and audio are stored in two separate files. The audio file does not contain any
time information. Only the video file contains the time information (it can be network time
stamp or local time stamp).


Frames needs to be stored in frame decode order or sequence number order.

*********************************************
Video File Format:
            Header
            Video_Data
            Index_Table
*********************************************

Header: Contains information regarding the File:
{
      VID_Header_HdInfo
      VID_Header_NTSInfo
      VID_Header_IndxInfo
      VID_Header_CodecInfo
      Padding
}

typedef struct {
      char        FileType[8];
      UInt32      HeaderSize;
      UInt32      VersionOfFileFormat;
      UInt32      PadSize;
      UInt32      nframe;
}VID_Header_HdInfo;

typedef struct {
      UInt32      nIndex;
      UInt32      IOffset;
}VID_Header_IndxInfo;

typedef struct {
      UInt32      StartVideoNTS;
      UInt32      StartAudioNTS;
}VID_Header_NTSInfo;

typedef struct {
      UInt32      profile;
      UInt32      level;
}VID_Header_H263CodecConfig;

typedef union {
      VID_Header_H263CodecConfig   h263;

}VID_Header_CodecConfig_Union;

typedef struct {
      char        VideoCodecName[8];
      UInt32      FrameRate;
      UInt32      BitRate;
```

```
      UInt32      ImageWidth;
      UInt32      ImageHeight;
      UInt32      SizeOfCodecConfig;
      VID_Header_CodecConfig_Union CodecConfig;
}VID_Header_CodecInfo;


00000000         CHAR FileType[8]= "VIDEO   ". Indicates that it is a Video
                 File.
00000008         UINT32 Size_of_the_Header: Total Header size in bytes
                 including the Size_of_the_Header and Padding
0000000C         UINT32 Version_of_File_Format: 16.16 Format, "XX.YY" XX=Major
                 Number, YY=Minor Number
00000010         UINT32 PADsize: To make sure Video Data can start on 4 byte
                 boundaries Add some padding. 0 means no Pad
00000014         UINT32 nframe: Number of Video Frames in the File
00000018         UINT32 StartVideoNTS: First Video packet Network or System
                 Time Stamp
0000001C         UINT32 StartAudioNTS: Video and Audio TSs are converted to
                 either Network or System TS when they arrive and stored. This
                 is to maintain the synchronization between Video and Audio
00000020         UINT32 nindex: Number of Frames in Index Table, 0 means there
                 is no Index_Table and ignore the offset
00000024         UINT32 IOffset: Offset for the Index_Table: Offset in 32 bit
                 from the start of the file
00000028         CHAR videocodecname[8]: NULL Terminated Video Codec Name. For
                 MPEG-4 "mp4v", for H263 "s263"
00000030         UINT32 FrameRate: (16.16 Format), From this we can deduct
                 frame buffering required for a given Task rate
00000034         UINT32 Bitrate: Bitrate of the Video Stream
00000038         UINT32 Imagewidth:
0000003C         UINT32 ImageHeight: These are used by the decoders to figure
                 out buffering requirements without scanning the compressed data.
00000040         UINT32 size of VideoCodecConfig :siz
00000044         UNION VideoCodecConfig: H.263/MPEG-4 Simple profile
00000044+siz     CHAR PAD[PADsize]:
**********************************************


**********************************************
Video_Data:
      Video_frame data[nframe];
**********************************************

Each Video_frame:
      UINT32 Frame_Size: Frame Size in bytes including (Frame_Size +
                         Packet_Count + npad + RTPPacket + Pad_data)

UINT32 Packet_Count: Number of Packets in the Current Frame.

UINT32 npad: Size of Pad; To make Frames as 4 byte boundaries we need
                  some padding. Indicates the amount of padding in
                  bytes. 0 means no pad.

RTPPacket packets[Packet Count]:

CHAR Pad_data[npad];
**********************************************


Each RTPPacket:

      UINT32 Packet_Size: Packet Size including the Header.

      UINT32 Packet_Arrival_Time_offset: With respect to the Video Arrival TS.

      UINT32 Video_RTP_Header: First 4 bytes of RTP Header
```

```
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    |V=2|P|X| CC |M|  PT      |       sequence number         |
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    |                    timestamp                                  |
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    Sequence number is stored in the file, to make sure that we
                    present the packets the same way when we play back, i.e., if
                    there is a packet loss and gap in the sequence numbers, while
                    playback the same will be maintained.

      UINT32 Video_RTP_Timestamp: Absolute Timestamp of the Video RTP Header
                    While playback these absolute values need to be converted to
                    relative values and added to the video current sequence
                    number and time stamp. Payload_Type needs to be replaced with
                    the payload_Type negotiated.

      CHAR Raw_Video_Data [Packet Size - 16]: Including Specific Playload
                    Header.
*********************************************


Index_Table: Contains the I Frames offset in the file.
      UINT32 frame_offset[nindex]: Offset of I frame from the start of the
                                   file.
```

5 Proprietary Native Audio File Format

This section describes the proprietary native audio file format.

```
Version of this File: 0.8

FILE_FORMAT:
AUDIO_HEADER
AUDIO_CODEC_HEADER
SAMPLE 1
SAMPLE 2
SAMPLE 3
  :
  :
SAMPLE N

AUDIO_HEADER:
typedef struct {
   char   AudioFileType[8];
   UInt32 SizeoftheHeader;
   UInt32 VersionOfFileForamt;
   UInt32 padSize;      //Number of extra bytes at the end of header to make it word aligned
   UInt32 nframe;
  }AUD_HeadInfo;

AudioFileType[]={'A','U','D','I','O',' ',' ',' '};
VersionOfFileFormat = 0x00000008;
SizeoftheHeader = sizeof(AUD_HeadInfo);

typedef struct {
   UInt32 Size;    //Size of this structure.
```

```
   UInt32 Coding;
   UInt32 SampleRate;
   UInt32 BitsPerSample;
  }AUD_CODEC_HEADER;              //Same as the MM_AUDIO_CODEC except version

SAMPLE:
         SAMPLE HEADER
         PayLoad

SAMPLE HEADER:
typedef struct {
   UInt32  size;   //includes payLoad + size of this structure
   UInt32  CoderType;
   UInt32  TimeStamp;
   UInt32  RefLocalTime;
   UInt16  SeqNo;
   UInt16  m;        //marker Bit
  }mm_aud_native_sample;

/*

 Marker Bit is set according to RFC

  The RTP header marker bit (M) SHALL be set to 1 if the first frame-
     block carried in the packet contains a speech frame which is the
     first in a talkspurt.  For all other packets the marker bit SHALL be
     set to zero (M=0).

  The coder type allows us to record RFC2833 or other Coder Packets in the same file.

 Seq Numbers are required so that we can record missed packets.
 TimeStamp and RefLocalTime are required to capture the jumps in TS.
*/
```

Publication Date: May 2007

Document Number: 05-2453-003

Intel
1515 Route 10
Parsippany, NJ 07054

For **Technical Support**, visit the Intel Telecom Support Resources website at:
http://developer.intel.com/design/telecom/support

For **Products and Services Information**, visit the Intel Telecom and Compute Products website at:
http://www.intel.com/design/network/products/telecom

For **Sales Offices** and other contact information, visit the Buy Telecom Products page at:
http://www.intel.com/buy/networking/telecom.htm