Dialogic
Making Innovation Thrive™

Dialogic® Multimedia File Conversion Tools

User Guide

February 2010

05-2453-009

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 Dialogic® Multimedia API, which provides the ability to record and play back digitized multimedia (audio and video) to support video services in application programs.

Applicability

This document may be applicable to releases of Dialogic® Multimedia Software for AdvancedTCA as well as Dialogic® Host Media Processing (HMP) Software.

How to Use this Publication

The information in this publication is organized as follows:

- Section 1, "Product Description", on page 5 – provides a brief description of the Dialogic® Multimedia File Conversion Tools, where to obtain them, and the proprietary multimedia format.
- Section 2, "The mmconvert Utility", on page 6 – provides detailed information for using the mmconvert utility.
- Section 3, "The hmp3gp Utility", on page 8 – provides detailed information for using the hmp3gp utility.
- Section 4, "Proprietary Video File Format", on page 12 – describes details of the proprietary video file format.
- Section 5, "Proprietary Native Audio File Format", on page 16 – 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 Dialogic® Multimedia API, see the following documents in the bookshelf for your product release:

- For information on Dialogic® Multimedia API features, an overview of video technology, guidelines for developing applications using the Dialogic® Multimedia API, in addtion to reference information on all functions, parameters, data structures, values, events, and error codes, see the *Dialogic® Multimedia API Programming Guide and Library Reference*.
- For information about the Dialogic® Multimedia API demo, see the *Dialogic® Multimedia 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
05-2453-009	February 2010	About This Publication chapter: Updated the Applicability section and the Related Information section. Features of the hmp3gp Utility section: Added H.264 support. Proprietary Video File Format section: Added H.264-specific header information.
05-2453-008	May 2008	How to Obtain the Tools section: Updated web site information. Features of the hmp3gp Utility section: Added several new features including WAV file format, extended reporting information, raw video, converting among proprietary file versions. Added support for H.263+. hmp3gp Command Line Interface section: Updated Command Line Parameters and parameter table. Expanded description of -d parameters. Added -d3, -i[sv], -h. Updated Examples. Proprietary Video File Format section: Updated for version 10. Proprietary Native Audio File Format section: Updated for version 10.
05-2453-007	December 2007	Features of the hmp3gp Utility section: Updated description to include option to convert proprietary audio file format to linear PCM audio file format. hmp3gp Command Line Interface section: Added -d 2 option to convert proprietary audio file to linear PCM audio file format.
05-2453-006	November 2007	Product Description section: Added note stating that the mmconvert utility is no longer recommended for use. The mmconvert Utility section: Added note stating that this utility is no longer recommended for use. Features of the hmp3gp Utility section: Updated video track information to include MPEG-4. Added 3GPP Release 5. hmp3gp Command Line Interface section: Added -i parameter. Changed g711b option to g711u option (correction). Added information about MPEG-4 and H.263. Added usage example for -i parameter. Proprietary Video File Format section: Updated for MPEG-4.

Document No.	Publication Date	Description of Revisions
05-2453-005	September 2007	Made global changes to reflect Dialogic brand. How to Obtain the Tools section: Updated web site information. hmp3gp Command Line Interface section: Added g711a, g711u options for -a parameter.
05-2453-004	August 2007	hmp3gp Command Line Interface section: Added g723_5300, g723_6300, g726, g729a options for -a parameter.
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 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.
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 *Dialogic®* 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 Dialogic® Multimedia File Conversion Tools Provided

This document provides information on the following Dialogic® 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 6.

Note: The mmconvert utility is no longer recommended; it is supported for H.263 video and linear PCM audio only. Instead, use the hmp3gp utility; see Section 3, "The hmp3gp Utility", on page 8.

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 8.

1.2 How to Obtain the Tools

For Dialogic® Host Media Processing (HMP) Software users on the Linux operating system, the conversion tools can be downloaded from the following web site. Check this web site periodically for any updates to the conversion tools and their capabilities, and for corresponding updates to this document:

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

For Dialogic® Host Media Processing (HMP) Software users on the Windows® operating system, the conversion tools can be downloaded from the following web site. Check this web site periodically for any updates to the conversion tools and their capabilities, and for corresponding updates to this document:

http://www.dialogic.com/support/helpweb/hmp/hmpwin/hmp30/omf/default.htm

For Dialogic® Multimedia Platform for AdvancedTCA users, the conversion tools can be downloaded from the following web site. Check this web site periodically for any updates to the conversion tools and their capabilities, and for corresponding updates to this document:

http://www.dialogic.com/support/helpweb/mmpatca/omf/default.htm

1.3 Proprietary Multimedia Format

The proprietary multimedia format is used by the Dialogic® 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 12.
- **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 16.

2 The mmconvert Utility

Note: The mmconvert utility is no longer recommended; it is supported for H.263 video and linear PCM audio only. Instead, use the hmp3gp utility; see Section 3, “The hmp3gp Utility”, on page 8.

This section provides detailed information about 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

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 16.
-v	video-version	specifies version of proprietary video file format used for output. Set to version mm07.
-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 about the hmp3gp utility. The following topics are covered:

3.1 Features of the hmp3gp Utility

The hmp3gp utility has the following capabilities:

- converts multimedia data in either direction between the following file formats:
  - 3rd Generation Partnership Project (3GPP) Release 4/Release 5 file format (.3gp)
  - proprietary multimedia format using an audio and a video file
- converts a proprietary audio file format to a linear PCM audio file format or to a WAV file format (reverse direction is not supported)
- retrieves extended information about proprietary multimedia files
- extracts raw video from a proprietary file by removing proprietary header information from the video file
- converts from one proprietary file version to another as needed (such as to gain support for a new feature)

- for proprietary video files: version 7 is the original version that supports H.263; version 8 adds MPEG-4; version 10 adds H.263+, DVR controls, and backwards compatibility with previous versions; version 10 also supports H.264.
- for proprietary audio files: version 8 is the original version; version 10 adds DVR controls and backward compability with previous versions.

The 3GP file contains two tracks: (1) a video track with H.263, H.263+ , H.264, or MPEG-4 video data, and (2) 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 utility performs CPU-intensive tasks and should only be used when sufficient CPU capacity is available and it will not affect other operations on the system. For example, it 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
        -i[sv] <proprietaryFile> [outputFile]
        -r <proprietaryFile> <outputFile>
        -d0 <inputVideoFile> <inputAudioFile> <output3gpFile>
        -d1 [options]<outputVideoFile> <outputAudioFile> <input3gpFile>
        -d2 [options]<inputAudioFile> <outputAudioFile>
        -d3 [options]<inputProprietaryFile> <outputProprietaryFile>
        -h <option>
```

Table 2. hmp3gp Parameters

Parameter	Options	Description
-i[sv]	[outputFile]	Returns information about a proprietary audio or video file. Valid values are: • -i – returns basic information about the proprietary file • -is – returns headers for each sample (useful for checking frame timestamps) • -iv – returns headers and raw sample data Optionally specify an output file to store the results. **Note:** This parameter is not supported for linear PCM file (mm07), as there is no file header on a linear file.
-d*n*		Specifies direction or destination of conversion between the specified files, where '*n*' represents: • 0 – converts proprietary video file to 3GP format (default) • 1 – converts 3GP to proprietary multimedia format • 2 – converts proprietary audio file format to 8 kHz linear PCM format • 3 – converts proprietary file from one version to another
-d0		Converts proprietary video file to 3GP format.
-d1	-a<audioFileFormat) -vv<version> -va<version>	Converts 3GP to proprietary multimedia format. Valid options are: -a which specifies audio file format: • mm07 – linear PCM (128 kbps) proprietary audio file format (default) • mm08 – proprietary native audio file format for AMR. For details, see Section 5, "Proprietary Native Audio File Format", on page 16. • g723_5300 – proprietary native audio file format for G.723 5.3 kbps • g723_6300 – proprietary native audio file format for G.723 6.3 kbps • g726 – proprietary native audio file format for G.726 • g729a – proprietary native audio file format for G.729a • g711a – proprietary native audio file format for G.711 a-law -vv which specifies the version of the video output file: • 0 – use the latest version (currently version 10) (default). For details, see Section 4, "Proprietary Video File Format", on page 12. • 7 and higher – use the version specified (7 or higher) **Note:** Versions 1-6 and 9 are not in use and are not valid. -va which specifies the version of the audio output file: • 0 – use the latest version (currently version 10) (default) • 8 and higher – use the version specified (8 or higher) **Note:** Versions 1-7 and 9 are not in use and are not valid.
-d2	-o<audioFileFormat>	Converts proprietary audio to other audio file format. Valid option is: -o which specifies audio file format: • mm07 – linear PCM (128 kbps) proprietary audio file format (default) • WAV – WAV PCM format

Table 2. hmp3gp Parameters (Continued)

Parameter	Options	Description
-d3	-v<version>	Converts proprietary file from one version to another. Valid option is: -v which specifies the version of the audio or video output file • 0 – convert to the latest version for this file type (currently version 10) (default) • 7 and higher – convert to the version specified (7 or higher) **Note:** Versions 1-6 and 9 are not in use and are not valid.
-r		Extracts raw data from a proprietary video file to an output file. The output file name extension will be added based on the input file. If no output file is specified, the proprietary input file name is used and an extension is added.
-h		Returns help about a parameter. For example: -hi returns information on the -i parameter -hd3 returns information on the -d3 parameter

Example 1: Convert to 3GP Format (-d0)

This example shows how to convert a proprietary format video and audio file to a 3GP format file.

```
hmp3gp -d0 <inputVideoFile> <inputAudioFile> <output3gpFile>
hmp3gp -d0 ver10conv.263.vid ver10conv.amr.aud output.3gp
```

Example 2: Convert to Proprietary Format (-d1)

This example shows how to convert a 3GP format file to proprietary format video and audio files.

```
hmp3gp -d1 <outputVideoFile> <outputAudioFile> <input3gpFile>
hmp3gp -d1 -amm08 test.mp4.vid test.amr.aud mpr.amr.3gp
```

Example 3: Convert to Linear PCM or WAV (-d2)

This example shows how to convert a proprietary audio file to linear PCM or WAV formats.

```
hmp3gp -d2 <inputAudioFile> <outputAudioFile>
hmp3gp -d2 test.amr.aud output.pcm
hmp3gp -d2 -owav test.amr.aud output.wav
```

Example 4: Convert Between File Versions (-d3)

This example shows how to convert a proprietary video file from version 8 to version 10.

```
hmp3gp -d3 <inputProprietaryFile> <outputProprietaryFile>
hmp3gp -d3 -v10 ver8.263.vid ver10conv.263.vid
```

Example 5: Return Information (-i)

This example shows how to retrieve information about a proprietary video file and display it on the screen.

```
hmp3gp -i <proprietaryFile>
hmp3gp -i v10conv.263.vid
```

The output is shown here:

```
PROP_Header_Info:
  sizeof():                 24
  FileType:                 VIDEO
  Size:                     92
  Version:                  0x000A (Video Version 10)
  PadSize:                  0
  nFrames:                  1280

VID_Header_NTSInfo:
  sizeof():                 8
  StartVideoNTS:            0
  StartAudioNTS:            0

VID_Header_IndxInfo:
  sizeof():                 8
  nIndex:                   9
  IOffset:                  1078496

VID_Header_CodecInfo:
  sizeof():                 28
  VideoCodecName:           s263
  FrameRate:                0x001E0000 (16.16 format)
  BitRate:                  64 (bits / second)
  ImageWidth:               352
  ImageHeight:              288
  SizeOfCodecConfig:        8

VID_Header_H263CodecConfig:
  sizeof():                 8
  profile:                  0
  level:                    10

VID_FileInfo:
  sizeof():                 16
  MediaLengthMs:            42633
```

4 Proprietary Video File Format

This section describes the proprietary video file format.

```
Version of this File: 10
General: Video and Audio are in two different files.
Advantages of two files: Audio file can be played using non video channels. If a video clip
contains multiple audio languages/channels (English, Spanish, German, etc.), the application can
choose which one can be played.
```

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

***********************************************

Video File Format:

Field Name                    Size in bytes   Data type
---------------------------------------------------------------------
Proprietary header            24              PROP_Header_Info
NTS header                    8               VID_Header_NTSInfo
Index header                  8               VID_Header_IndxInfo
Codec info header             28              VID_Header_CodecInfo
Codec Config header           variable        VID_Header_*_CodecConfig
Generic file information      16              VID_FileInfo
Padding                       variable        UInt8 [PadSize]

Begin Frame #1
  Frame header                12              VID_VideoFrame
  Extended data size          4               VID_ExtDataSize
  Extended frame data         4               VID_FrameInfo

  Begin Packet #1
    Packet header             16              VID_RTPPacket
    Extended data size        4               VID_ExtDataSize
    Extended packet data      0               none
    Packet payload            variable        UInt8 [payload bytes]
  End Packet #1
  ...repeat Packet for each packet up to PacketCount

  Frame padding               variable        UInt8 [FramePad]
End Frame #1
... repeat Frame for each frame up to FrameCount

Index table                   12 * nIndex     VID_IframeIndx [nIndex]
***********************************************

/*
 * Proprietary file header
 * Includes generic information about the contents of the file.
 *
 * Size field is the size of the entire header.
 * I.e. Size is equal to the offset of the first frame.
 */
typedef struct {
        char      FileType[PROP_FILE_TYPE_SIZE];
           /* magic string (MMRSC_AUD_FILE_TYPE or VID_FILE_TYPE) */
        UInt32    Size;
           /* Size of entire header (AUD_HEADSIZE or VID_HEADSIZE plus variable header size) */
        UInt32    VersionOfFileFormat;
           /* Version of this file format */
        UInt32    PadSize;
           /* Padding, in bytes */
        UInt32    nFrames;
           /* Number of frames contained in this file */
} PROP_Header_Info, VID_Header_HdInfo, AUD_HeadInfo;

/*
 * Extended data size
 *
 * Contains the amount of extended data to follow this structure.
 * Does not include the size of itself.
 */

typedef struct {
      UInt32      ExtDataSize;
} VID_ExtDataSize, AUD_ExtDataSize;
```

```
/*
 * File information structure
 */
typedef struct {
      UInt32 MediaLengthMs;   /* Length of the Video/Audio in this file in MS */
      UInt32 RFU[3];          /* RFU space to add to this structure without changing its size */
} VID_FileInfo, AUD_FileInfo;

/*
 * Index table info
 */
typedef struct {
        UInt32    nIndex;
        UInt32    IOffset;
}VID_Header_IndxInfo;

/*
 * Audio and Video offset info
 */
typedef struct {
        UInt32    StartVideoNTS;
        UInt32    StartAudioNTS;
}VID_Header_NTSInfo;

/*
 * H.263 specific header
 */
typedef struct {
        UInt32    profile;
        UInt32    level;
}VID_Header_H263CodecConfig;

/*
 * H.264 specific header
 */
typedef struct {
        UInt32    profile;
        UInt32    profile_iop;
        UInt32    level;
            UInt8     dciBuffer[350];
            UInt16    dciSize;
}VID_Header_H264CodecConfig;

/*
 * MPEG4 specific header
 */
typedef struct {
        UInt32    profilelevelID;
            UInt8     dciBuffer[350];
            UInt16    dciSize;
}VID_Header_MPEG4CodecConfig;

/*
 * Codec info
 *
 * The VideoCodecName field determines which variable header(s)
 * will follow the fixed headers.
 *
 * SizeOfCodecConfig describes the length of the variable header(s)
 */
typedef struct {
        char      VideoCodecName[8];
        UInt32    FrameRate;
            UInt32    BitRate;
        UInt32    ImageWidth;
        UInt32    ImageHeight;
        UInt32    SizeOfCodecConfig;
}VID_Header_CodecInfo;
/*

 * Video Frame header
 *
 * FrameSize is equal to the ENTIRE frame size, including this structure,
 *   all headers, packets and payloads.
 */

typedef struct {
        UInt32    FrameSize;
        UInt32    PacketCount;
        UInt32    FramePad;
}VID_VideoFrame;

typedef struct {
        UInt32    FrameSize;
        UInt32    PacketCount;
        UInt32    is_Iframe;
        UInt32    FramePad;
}VID_VideoFrameV2;

/*
 * Video Frame info
 *
 * Extended frame info.  A bitmask that may contain zero or more of the following flags:
 *
 * VID_IS_IFRAME:       This frame is an I-frame
 *
 */
typedef struct {
      UInt32 frmFlags;
} VID_FrameInfo;

// Frame info bits
#define VID_IS_IFRAME   0x00000001

/*
 * Video I-frame index info
 */
typedef struct {
      UInt32    offset;             /* offset from start of file to the I-frame */
      UInt32    timestamp;          /* RTP timestamp of the frame */
      UInt32    framenum;           /* sequential number of this frame in the file */
} VID_IframeIndx;

/*
 * Video Packet header
 *
 * PacketSize is equal to the ENTIRE packet size, including this structure
 *   and payload
 */
typedef struct {
        UInt32    PacketSize;
        Int32     PktArriavalTimeOffset;
        UInt32    PktRTPHeader1;          // First 4 bytes of RTP packet
        UInt32    PktRTPHeader2;          // Packet timestamp
}VID_RTPPacket;
```

5 Proprietary Native Audio File Format

This section describes the proprietary native audio file format.

```
/********************************************************************

MMRSC_AUD_FILE_VER_10 file structure:

Field Name                    Size in bytes   Data type
---------------------------------------------------------------------
Proprietary header            24              PROP_Header_Info
Codec header                  16              AUD_Codec_Header
Index header                  12              AUD_Header_IndxInfo
Generic file information      16              AUD_FileInfo
Padding                       variable        UInt8 [PadSize]

Begin Frame #1
  Frame header                20              AUD_AudFrame_Native
  Extended data size          4               AUD_ExtDataSize
  Extended frame data         0               none
  Frame payload               variable        UInt8 [Size - sizeof(AUD_AudFrame_Native)]
End Frame #1

... repeat Frame for each frame up to FrameCount

Index table                   12 * nIndex     AUD_Indx [nIndex]

********************************************************************/

typedef struct {
        char      FileType[PROP_FILE_TYPE_SIZE];
            /* magic string (MMRSC_AUD_FILE_TYPE or VID_FILE_TYPE) */
        UInt32    Size;
            /* Size of entire header (AUD_HEADSIZE or VID_HEADSIZE plus variable header size) */
        UInt32    VersionOfFileFormat;          /* Version of this file format */
        UInt32    PadSize;                      /* Padding, in bytes */
        UInt32    nFrames;                      /* Number of frames contained in this file */
} PROP_Header_Info, VID_Header_HdInfo, AUD_HeadInfo;

/*
 * Extended data size
 *
 * Contains the amount of extended data to follow this structure.
 * Does not include the size of itself.
 */

typedef struct {
      UInt32      ExtDataSize;
} VID_ExtDataSize, AUD_ExtDataSize;

/*
 * File information structure
 */

typedef struct {
      UInt32 MediaLengthMs;   /* Length of the Video/Audio in this file in MS */
      UInt32 RFU[3];          /* RFU space to add to this structure without changing its size */
} VID_FileInfo, AUD_FileInfo;
```

```
/****************** audio file header *******************/
/*
 * Codec info
 */

typedef struct {
        UInt32 Size;    //Size of this structure.
        UInt32 Coding;
        UInt32 SampleRate;
        UInt32 BitsPerSample;
}AUD_Codec_Header;

/*
 * Index info
 *
 * Only in Version 9 and later
 */

typedef struct {

      UInt32    nIndex;    /* number of elements in index table */
      UInt32    offset;    /* offset from start of file to index table */
      UInt32    interval;  /* number of packets between successive indices */
}AUD_Header_IndxInfo;

/****************** END: audio file header *******************/
/*

 * Header for each audio RTP packet
 *
 * Size field is equal to the ENTIRE size of the packet,
 *   including the size of this structure and the payload.
 */

typedef struct {
        UInt32 Size; //includes payload + size of this structure
        UInt32 CoderType;
        UInt32 TimeStamp;
        UInt32 RefLocalTime;
        UInt16 SeqNo;
        UInt16 m;         //marker Bit

}AUD_AudFrame_Native;


typedef struct {
      UInt32 offset;      /* offset from start of file to the packet */
      UInt32 timestamp;   /* timestamp of the packet referenced by this index */
      UInt32 blocknum;    /* sequential number of this block in the file */
} AUD_Indx;
```

Copyright and Legal Notices

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries (“Dialogic”). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic’s legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Dialogic, Dialogic Pro, Brooktrout, Diva, Diva ISDN, Making Innovation Thrive, Video is the New Voice, Diastar, Cantata, TruFax, SwitchKit, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, TrustedVideo, Exnet, EXS, Connecting to Growth, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic’s legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic’s trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Publication Date: February 2010

Document Number: 05-2453-009